File No. S360-35
Form C28-6501-2

IBM Systems Reference Library

IBM 7090/7094 Support Package for IBM System/360

The IBM 7090/7094 Support Package for the IBM System/360 consists of three programs designed to permit the assembly, testing, and execution on an IBM 709, 7090, 7094, or 7094 II Data Processing System of programs written for an IBM System/360. The three component programs are a 7090 Assembly Program, a 7090 Simulator Program, and a 1401 Input Program.

The Support Package can process all System/360 assembler language and machine language programs that are not specifically dependent on input/output timing considerations. It accepts the machine instruction and assembler instruction mnemonic codes of the IBM System/360 Special Support Basic Assembler Language and simulates the standard, scientific, commercial, universal, storage protection, and direct control instruction sets of the System/360. It simulates most input/output operations and most interruption procedures of the System/360. Simulation of up to 64K bytes of System/360 main storage is allowed. With certain limitations, the Support Package simulates the operations of the 1052 Printer-Keyboard, 1402 Card Read Punch, 1403 Printer, 1442 Card Read Punch, 1443 Printer, 2311 Disk Storage Drive, and the 2401, 2402, 2403, and 2404 Magnetic Tape Units. In addition, it provides extensive facilities for use in detecting and tracing errors in object programs.

The assembly and Simulator Programs operate under a supervisor program called in by the 7090/7094 Operating System Monitor (IBSYS).

PREFACE

This publication contains specifications for use of the IBM 7090/7094 Support Package for IBM System/360. The Support Package consists of two 7090/7094 programs--an Assembly Program and a Simulator--and a 1401 Input Program. These programs are designed to permit programs written for a System/360 to be assembled, tested, and executed on an IBM 709/7090/7094/7094 II. Wherever the 7094 is referred to in this manual, the 7094 II also is implied.

It is assumed that the reader of this publication is familiar with the System/360 Principles of Operation manual and with the System/360 Special Support Basic Assembler Language (usually referred to herein as the Basic Assembler Language). It also is assumed that he has some knowledge of the 7090/7094 Operating System, especially its Basic Monitor (IBSYS).

The following publications contain information that may be helpful in using the Support Package:

IBM System/360 Special Support Basic Assembler Language, Form C28-6503 (contains source specifications)

IBM System/360 Operating System Assembler Language, Form C28-6514 (contains additional source specifications)

IBM System/360 Principles of Operation, Form A22-6821 (machine reference manual)

IBM 7090/7094 IBSYS Operating System, System Monitor (IBSYS), Form C28-6248 (contains information on the use of the 7090 System Monitor)

IBM 7090/7094 IBSYS Operating System, Operator's Guide, Form C28-6355 (contains operating instructions for the 7090 Operating System)

IBM 709 Data Processing System, Form A22-6536 (machine reference manual)

IBM 7090 Data Processing System, Principles of Operation, Form A22-6528 (machine reference manual)

IBM 7094 Data Processing System, Principles of Operation, Form A22-6703 (machine reference manual)

IBM 7094 II Data Processing System, Form A22-6760 (machine reference manual)

IBM 7090/7094 Utility Routines for IBM 1301 Disk Storage, Form J28-6223 (contains specifications and operating instructions for 1301 utilities)

MAJOR REVISION (November 1964)

This publication is a major revision of Form C28-6501-1, which is now obsolete. This publication contains new material on the simulation of an additional input/output device (the 2311 Disk Storage Drive), an additional control card (the OUTPUT card), the Read Backwards feature on the 2401, 2402, 2403, and 2404 Magnetic Tape Units, simulation of thc Card Image feature of the 1442 Card Read Punch, and the Column Binary feature of the 1402 Card Read Punch. Changes in the text are identified by a vertical line to the left of the revised material.

This publication was prepared for production using an IBM computer to update the text and to control the page and line format. Page impressions for photo-offset printing were obtained from an IBM 1403 Printer with a 120-character chain containing upper and lower case letters, special characters, and numerals.

Copies of this and other IBM publications can be obtained through IBM Branch Offices.

A form has been provided at the back of this publication for readers' comments. If the form has been detached, comments may be directed to an IBM Systems Engineer or addressed to the IBM Corporation, Programming Systems Publications, Dept. D58, PO Box 390, Poughkeepsie, N.Y. 12602.

THE IBM 7090/7094 SUPPORT PACKAGE FOR THE IBM SYSTEM/360

The IBM 7090/7094 Support Package for the IBM System/360 (referred to herein as the 7090 Support Package) consists of three programs designed to permit programs written for a System/360 to be assembled, tested, and executed on an IBM 709, 7090, 7094, or 7094 II. The 7090 Assembly Program will accept source programs written in System/360 Basic Assembler Language and will produce object programs in System/360 machine language, ready for execution. Such programs may be executed either on an actual System/360 or by further use of the 7090 Support Package. The 7090 Simulator Program will accept the output of the Assembly Program, or a System/360 machine-language program prepared by other means, and will simulate its execution. Input to both 7090 programs is prepared originally in punched card form and is converted to magnetic tape format by the 1401 Input Program.

In general, the Support Package can process all System/360 assembler language or machine language programs that are not specifically dependent on input/output timing considerations, subject to certain restrictions described in this publication. The Assembly Program can accept all of the machine instruction mnemonic codes and assembler instruction mnemonic codes of the Basic Assembler Language, and the Simulator can process the standard, scientific, commercial, universal, storage protection, and direct control instruction sets of the System/360; it also simulates most of the input/output operations and interruption procedures. Simulation of up to 64K bytes of System/360 main storage is allowed. With certain limitations, the Support Package simulates the operations of the 1052 Printer-Keyboard, 1402 Card Read Punch, 1403 Printer, 1442 Card Read Punch, 1443 Printer, 2311 Disk Storage Drive, and the 2401, 2402, 2403, and 2404 Magnetic Tape Units. In addition, it provides extensive facilities to aid the programmer in detecting and tracing errors in an object program.

The Assembly and Simulator Programs operate as one system under a supervisor program called in by the 7090/7094 Operating System Monitor (IBSYS).

SCOPE OF THIS PUBLICATION

This publication is concerned only with the 7090 Support Package itself, not with the rules that govern the writing of programs in the Basic Assembler Language or in System/360 machine language, or with the operating procedures of an actual System/360. Specifically, this manual deals with the following:

1. The range of simulation to be expected and the restrictions imposed on it.

2. The format and structure of input and output.

3. The control cards required in calling in the facilities of the Support Package.

4. Certain aspects of 7090 and 1401 machine operation that pertain specifically to the Support Package.

The 7090 Operating System Monitor (IBSYS) is generally referred to in this publication as the "7090 System Monitor." For convenience, certain aspects of the 7090 System Monitor are described in this text, and System Monitor unit names, such as SYSOU1, SYSPP1, and the like, are identified in Appendix A.

7090 AND 1401 MACHINE CONFIGURATION

The 7090 Support Package requires the use of an IBM 709 Data Processing System with data channel traps, or an IBM 7090 or 7094 Data Processing System, with the following minimum configuration:

1. 32K positions of core storage

2. One IBM 711 Card Reader

3. One IBM 716 Printer

4. One input/output unit containing the Assembly and Simulator Programs, as part of the System Library (SYSLB1). This may be either of the following:

   a. One IBM 729 Magnetic Tape Unit

   b. One IBM 7909 Data Channel with either (a) one IBM 7640 Hypertape Control Unit and one IBM 7340 Hypertape Drive or (b) one IBM 7631 File Control with the Cylinder Mode Feature and selected cylinders of one IBM 1301 Disk Storage Unit

5. IBM 729 Magnetic Tape Units, as follows:

   a. For assembly only, five units

   b. For simulation only, two units

   c. For both assembly and simulation, six units

   d. Certain System/360 input/output units may be simulated on the required 729 magnetic tape units; additional 729 units may be used in simulating other System/360 input/output units (see Appendix A). These 729 units must be on Channel A and/or B

An IBM 1401 Data Processing System with the following minimum configuration is also required:

1. 4K positions of core storage

2. Advanced Programming Feature

3. Column Binary Feature

4. High-Low-Equal Compare Feature

5. One IBM 729 or one IBM 7330 Magnetic Tape Unit

6. One IBM 1402 Card Read Punch

7. One IBM 1403 Printer with 132 print positions

The 7090 Assembly Program is used to produce object programs in System/360 machine language from source programs in the System/360 Basic Assembler Language. The 7090 Assembly Program, however, has extended facilities, some taken from the System/360 Operating System Assembler. (See "Additional Facilities" on this page.)

The object programs in System/360 machine language are intended for execution by the 7090 Simulator Program but may also be run on an actual System/360. The original source program must be punched in cards for processing by the 1401 Input Program, which writes it on magnetic tape for input to the Assembly Program. The latter then creates a machine-language object program and writes it on magnetic tape in two forms. One, on the System Output Unit (SYSOU1), is used to prepare a listing. The other, on the System Peripheral Punch Unit (SYSPP1), is used to punch a card deck in column binary form; this deck may then be converted by the 1401 Input Program into a tape for input to the Simulator Program. It is also possible to specify that a program be simulated immediately upon assembly; in that case, the assembled output is also written on a third magnetic tape (SYSUT2) and that tape is used directly as input to the 7090 Simulator Program.

The source program must be written in the Basic Assembler Language according to the rules for that language as specified in the Basic Assembler Language publication. Exceptions are noted below under "Additional Facilities." The Assembly Program will accept and process all of the machine instruction mnemonic codes and all assembler instruction mnemonic codes of the Basic Assembler Language. However, only the 48 characters of Character Set H may be used in punching the source deck; other System/360 characters are not accepted. The characters of Character Set H are as follows:

A - Z
0 - 9
Blank
Special characters + - * / = . , $ ' ( )

Note: $ is not treated as an alphabetic character by the Support Package.

Five of these characters may be punched either in standard IBM card code or in IBM extended card code. However, only one type of card code will be accepted in any one program segment (see below under Output from the Assembly Program). The alternate codes are as follows:

CHARACTER	STANDARD IBM CARD CODE	IBM EXTENDED CARD CODE
+	12	12-6-8
(	0-4-8	12-5-8
)	12-4-8	11-5-8
=	3-8	6-8
'	4-8	5-8

ADDITIONAL FACILITIES

The 7090 Assembly Program implements an assembly language at a slightly higher level than System/360 Basic Assembler Language. These are the points of difference:

1. The location counter can reach a maximum value of 16,777,215 (2 to the 24th power minus 1), rather than 65,535 (2 to the 16th power minus 1).

2. Self-defining values have a maximum of 16,777,215, rather than the various limits imposed by the System/360 Basic Assembler Language.

3. Values associated with statement names have a maximum of 16,777,215, rather than 65,535.

4. Compound expressions may have more than three elements. Division (/) is allowed. Compound expressions may be used in all subfields of the variable field in the following statements: machine instructions, CCW's, USING, DROP, EQU, ORG, END.

5. The extended branch mnemonics specified in the System/360 Operating System Assembler Language are allowed.

6. DC types C and X have a maximum explicit length of 256 rather than 16. DC types P, Z, and S are allowed. Types P and Z are specified in the System/360 Operating System Assembler Language. The operation and variable fields of type S are written as "DC S(simply relocatable expression)." The expression is assembled as a properly aligned half-word specifying a base register (four high-order bits) and a displacement (twelve low-order bits). Neither duplication factor nor length modifier is allowed.

7. The assembler instructions USING and DROP may specify lists of registers, each item in the list separated from the next by a comma.

8. ICTL has one operand in the variable field, denoting the starting column, and this must be any decimal number from 1 through 40. An error in an ICTL card will terminate assembly.

9. The assembler instruction TITLE is implemented, as defined in the System/360 Operating System Assembler Language (Continuation cards not allowed).

10. The Assembly Program produces a cross-reference table of all symbols used during the assembly. References are by location counter value, and in the case of DC types A, references are given before alignment.

11. If the value of any absolute expression in the variable field of a machine instruction or CCW exceeds the maximum allowable (e.g., 4 bits for index field), the rightmost bits will be used and a warning flag placed in the listing.

INPUT DECK STRUCTURE

The source deck must be organized in the same manner as any deck intended for assembly by the assembler for an actual System/360, but also must be preceded by a TITLE card and optionally by COUNT and/or OUTPUT cards, and must be followed by an End-of-File card with the characters EOF in columns 2-4. The structure of the source deck is summarized in the list below, which is followed by a brief description of the special cards required by the 7090 Assembly Program:

1. TITLE card

2. COUNT card (optional)

3. OUTPUT card (optional)

4. Source program cards for assembly, beginning with START statement card (optional) and ending with END statement card

5. End-of-file card (see Appendix B)

Note: The END statement card referred to is one provided in the source program; it is not the same as the END card produced by the Assembly Program. The latter is generally referred to as a Load End card.

The formats of these cards are described in Appendix B.

TITLE Card

The TITLE card indicates to the 7090 Support Package supervisor the beginning of the source deck and initiates processing by the Assembly Program. The programmer also may specify on this card a title for the program, and this title field will be printed at the head of each page of the listing.

COUNT Card

The COUNT card is optional. When used, it should contain an approximate count of the number of cards in the source deck. If the COUNT card is not used, this number is assumed to be 1,000. The Assembly Program uses the count to minimize processing time by rewinding tapes at the optimum time. All of the cards in the source deck will be processed, regardless of the count specified.

OUTPUT Card

The OUTPUT card is optional. It may be used to suppress the program listing (except for those statements that contain errors), the cross reference table, and the binary card deck.

START and END Statement Cards

The START and END statement cards are standard for programs written for System/360 and are described in the Basic Assembler Language publication.

STRUCTURE OF THE ASSEMBLY PROGRAM

The Assembly Program consists of two principal phases, as described below.

Phase I

Phase I begins with procedures in which the TITLE,COUNT, and OUTPUT cards are processed and other initialization routines are performed. The source cards are then scanned, instruction codes are analyzed, and main storage is allocated for instructions, constants, work areas, etc. This process results in an "internal dictionary" in which are recorded, among other things, the storage locations associated with the symbols used in the program. A compressed form of internal text is produced from the original source language statements; in this form of the program, symbols are replaced by references to the internal dictionary. When this activity has been completed, any references to other programs are analyzed and an "external symbol dictionary" (described later in this section) is produced.

Phase II

In Phase II, the compressed internal text is processed to create the sequence of machine-language instructions and data that make up the object program. The "relocation list dictionary," which is used by the loader in relocating assembled addresses, is then produced. Following this, symbols are sorted and listed as required, together with their appropriate cross references.

OUTPUT FROM THE ASSEMBLY PROGRAM

The sequence of machine-language instructions that is produced by one execution of the Assembly Program is known as a "program segment," or simply, as a "segment." Such a segment may constitute an entire object program, or it may be part of a larger program. Each segment is produced in relocatable form. That is, the assembled addresses may be adjusted at load time by the addition or subtraction of a "relocation factor" to give new addresses for use at that time. (If the program is not relocated, the assembled addresses remain absolute.) The 7090 Support Package includes a loader to accomplish relocation, and it is assumed in this publication that the user will employ this loader. However, the user may employ a loading program in System/360 machine language, as explained in Section 4.

A program segment in source language may contain references to symbols that are used in other segments; in this case, it is assumed that all segments so referenced will be available together in main storage at object time. A symbol referred to in one segment but defined in another is called an "external symbol" in the segment containing the reference, but is called an "entry point" in the segment where it is defined. External symbols are identified in the source program by means of the EXTRN assembler instruction, while entry points are defined by the assembler instruction ENTRY. The method of coding these assembler instructions is explained in the Basic Assembler Language publication.

The Assembly Program will check for certain common coding errors in the source program, and post an error flag beside any offending statement in the assembled program listing. (See Appendix H for a list of error flags.)

Text Card (TXT)

The actual text of the object program is punched in "text cards," which are identified by the symbol TXT in columns 2-4. Such a card may contain as many as 56 bytes of machine-language text. It also specifies the assembled address of the first byte of text on the card. If the program is relocated, the relocation factor will be applied to each assembled address. For an object program to be relocatable, it must be preceded by External Symbol Dictionary (ESD) cards. If ESD cards are not present at load time, the assembled addresses will be treated as absolute.

External Symbol Dictionary Card (ESD)

At least one ESD card is produced in each execution of the Assembly Program. Such cards, which are identified by the symbol ESD in columns 2-4, are of three kinds, differing slightly in format. One contains the name, origin, and length of the segment. A second contains the names and addresses of any entry points to the segment. The third refers to the names of any external symbols referred to in the segment. ESD cards may be removed from the card deck before loading. If they are removed, the program will not be relocated.

Relocation List Dictionary Card (RLD)

These cards, identified by the symbol RLD in columns 2-4, specify the locations of any addresses in the current segment that will require relocation. Each item in the list contains a reference to the external or internal name to be used in determining the relocation factor, a reference to the segment containing the address to be relocated, and the assembled address. If the ESD cards for the segment are removed prior to loading the deck, the RLD cards will be ignored.

Load End Card (END)

The Load End card defines the end of the segment. It may also specify an entry point to the program.

Note: This card is not the same as the END card supplied by the programmer in the source program deck; that card is called an END statement card.

The detailed formats of these cards are given in Appendix B. Their use as input to the loader is further discussed in Section 3, which describes the structure of the input deck.

SECTION 2: 7090 SIMULATOR PROGRAM

The 7090 Simulator Program may be used to simulate, on the 7090, the execution by a System/360 of object programs in the machine language of the System/360. As part of the 7090 Support Package, it is particularly intended to accept the output of the 7090 Assembly Program; but it will also accept System/360 machine language prepared by other means, as long as the input is in proper format.

In general, the Simulator Program is not intended to simulate the exact manner in which a System/360 would execute a given machine instruction, but rather to produce the same result. Input/output operations can be carried out simultaneously with processing in the central processing unit, but no attempt is made to simulate timing relationships.

The program can simulate the standard, scientific, commercial, universal, storage protection, and direct control instruction sets of the System/360. It also simulates most input/output operations and most of the interruption procedures of the System/360.

The various simulation facilities may be called in by the use of control cards in the input deck and/or by using one or more of the extended forms of the Supervisor Call instruction that are provided for the purpose in the Simulator Program. The control cards are discussed in this text in connection with their related functions; the specific formats are given in detail in Appendix B. The Supervisor Call instruction is discussed primarily in connection with the dynamic dump facility and normal termination procedures (later in this section) and in Section 4; for convenience, it is usually referred to by its mnemonic symbol, SVC.

The Initial Program Loading procedure (IPL) may be simulated if specified by an IPL control card, as explained in Section 4.

The WAIT state is simulated by testing, for a limited time, for input/output interruptions, operator interruptions, and operator termination procedures. An on-line message then is printed that requests the operator to take appropriate action. The WAIT state is indicated when 7090 Sense Lights 2, 3, and 4 are off; these Lights simulate the Console Running Light of the System/360.

RANGE OF SIMULATION

CPU SIMULATION

The range of machine instructions to be simulated may be specified by the use of a CPUCF (CPU configuration) control card. This card specifies the storage size of the System/360 and the instruction sets available. (The actual format of the CPUCF card is given in Appendix B.)

INTERRUPTIONS

Most System/360 interruptions are simulated, as described in the following text.

Program Interruptions

All Program interruptions that would be executed by a System/360 are simulated. That is, the correct interruption bits are set in the program status word (PSW), the PSW is stored, and the new PSW is loaded. In addition, an appropriate message is written on the System Output Unit (SYSOU1) when any of the following Program interruptions occurs:

Operation
Privileged Operation
Execute
Protection
Addressing
Specification
Data

It should be noted, however, that only one interruption is provided for on the execution of any one machine instruction.

Supervisor Call Interruptions

Supervisor Call interruptions are simulated. However, Supervisor Call (SVC) instructions with certain identification codes are interpreted by the Support Package not as normal SVC instructions but as special requests to the Support Package. The uses of the SVC 1 and SVC 0 instructions are described later in this section in connection with calling in dynamic dump routines and terminating the execution of a program. Further information is contained in Section 4.

Input/Output Interruptions

The Input/Output interruptions are simulated as explained later in a separate subsection of this section titled "Simulation of Input/Output Interruptions."

Other Interruptions

Timer and machine error interruptions are not simulated.

INPUT/OUTPUT SIMULATION

Simulation of input/output operations is limited to providing accurate transmission of data to and from simulated System/360 main storage. Timing has not been considered, and any System/360 program that depends on the timing of input/output operations may fail.

Overlapping of 7090 input/output operations with simulation of the System/360 CPU is achieved by means of 7090 data channel trapping through the use of the Input/Output Executor (IOEX) of the 7090 System Monitor. The points in the System/360 program where input/output interruptions occur will depend only on the timing of the 7090 input/output operations, and will not necessarily correspond to the points where they would occur during actual operation on a System/360.

The input/output section of the 7090 Simulator Program permits the simulation of units attached to the multiplexor channel and to the selector channels. No standard input/output configuration is assumed. All units to be simulated must be defined by means of IOCF control cards, as described in Appendix B.

Following is a list of System/360 input/output units and the IBM 7090 units on which they can be simulated:

IBM SYSTEM/360 UNIT	IBM 7090 UNIT
1052 Printer-Keyboard	
Input	711 Card Reader
Output	716 Printer
Attention Key	Sense Switches 2 and 6
1402 Card Read Punch	729 Tape Units
1442 Card Read Punch	729 Tape Units
1403 Printer	729 Tape Units
1443 Printer	729 Tape Units
2311 Disk Storage Drive	1301 Disk Storage (or 7320 Drum Storage)
2401, 2402, 2403, and 2404 Tape Units	729 Tape Units

The simulation of each of these System/360 units, and of the System/360 console, is described in detail under "Simulation of Input/Output Devices."

Input/Output Instructions

All four input/output instructions (Start I/O, Test I/O, Halt I/O, and Test Channel) are simulated, but the following points should be noted regarding the Start I/O and Halt I/O instructions.

Start I/O: The 7090 Simulator Program does not simulate the channel programming error caused by the initial data address exceeding the addressing capacity of the System/360 model.

Halt I/O: Because of differences in the operating characteristics of the 7090 and the System/360, it is not possible to simulate the Halt I/O instruction with complete accuracy. Input/output simulation depends on the operation of the Input/Output Executor (IOEX), and under IOEX the 7090 instruction Reset Data Channel cannot be used with safety; once 7090 input/output operations are initiated, they must be allowed to come to a normal termination. Therefore, a Halt I/O instruction cannot take effect during an operation that involves data chaining through a list of channel command words. Consequently, a Halt I/O instruction can never cause the reading or writing of partial records, but only of complete or null records.

If the unit to be halted is on a selector channel that is busy, the Simulator will wait for the operation to be completed and the interruption stacked. If the unit to be halted is on a multiplexor channel, the Simulator will not wait for termination, as it is assumed that the control unit, if busy, is operating in Multiplex mode; the type of unit involved is immaterial.

Direct Control Feature

The Simulator processes the Write Direct and Read Direct instructions and the External interruption as described in the following text.

Write Direct Instruction: When a Write Direct instruction is encountered, the contents of the byte of storage to which it refers are displayed as the rightmost eight bits of the MQ register, while the contents of the I2 field of the instruction are displayed as the leftmost eight bits of the MQ. (The I2 field contains timing signals.) The 7090 then pauses to permit the operator to perform any special action the user may have specified. A message is given to the operator to set the keys as required and then to press START to continue.

Read Direct Instruction: When a Read Direct instructionis encountered, the 7090 pauses and a message is given to the operator to set the keys as required by the user, including key S, and to press START to continue. The leftmost eight bits of the MQ register will display the contents of the I2 field of the instruction, which contains timing signals. Setting key S cancels the Hold signal and causes one byte from keys 28-35 to be placed in main storage at the address indicated by the instruction. This is also recorded on the System Output (SYSOU1). Simulation of the Read Direct instruction is limited to accepting eight bits of data.

External Interruption: Sense Switches 4 and 6 are used to simulate External interruptions. If any one of keys 30-35 is on, the pattern indicated by those keys is stored in bits 26-31 of the old program status word. A new PSW is then loaded. The action required on the part of the operator is discussed later under "Operator Facilities."

Channel Command Words

The following channel commands are simulated insofar as they apply to the unit to which they are addressed:

Transfer in channel
Read
Write
Control
Sense

The treatment of command modifier bits varies with the particular type of input/output unit with which they are used. They are discussed in connection with the units to which they apply.

All flag bits are accepted by the Simulator and have their normal effect except that Program-Controlled interruptions (PCI) are subject to the restrictions described under "Simulation of Input/Output Interruptions."

Simulation of Input/Output Devices

System/360 Console

Certain features of the System/360 console are simulated by the 7090 features shown in the following table:

CONSOLE FEATURE	7090 FEATURE	REMARKS
Running Light	Sense Lights 2, 3, and 4	Lights will be on when there is no wait bit in the current PSW, off when the wait bit is present.
IPL Light	Sense Light 1	Light will be on during an IPL procedure, otherwise off.
Console Interruption	Sense Switches 3 and 6	

For information on operator action that may be required in connection with the above, see the subsection "Operator Facilities," later in this section.

1052 Printer-Keyboard

Input and output through the 1052 Printer-Keyboard are simulated separately, with input processed through the 711 Card Reader, and output through the 716 Printer. The 1052 Attention signal is simulated by Sense Switches 2 and 6.

Input: Messages for input must be punched in cards and entered through the 711 Card Reader. Card columns 1 through 70 may be used for text, while columns 71 and 72 must contain two decimal digits specifying the number of characters in the text. The text is considered to begin in column 1; thus, all blanks, including any that precede the first non-blank character, will be considered in the character count, and any characters in excess of the number specified will be ignored. Only the following 45 characters should be used:

A - Z
0 - 9
Blank
Special characters + - / = . , $ '

| NOTE: $ is not treated as an alphabetic character by the Support Package.

After the message has been read, it will appear on the on-line 716 Printer, preceded by the words CONSOLE INPUT. All input messages are assumed to be terminated by a carriage return; i.e., Console Printer output or further Console Keyboard input on the same printed line cannot be produced.

Output: Console Printer output is simulated on the 716 Printer, with each message preceded by the words CONSOLE OUTPUT. The maximum line length is 84 characters. Carriage control codes are treated as part of the output. Output is restricted to the 45 characters specified above for input.

1402 Card Read Punch

Complete simulation of a 1402 requires two tape units, one simulating card input, the other simulating card output. If either input or output, but not both, is to be simulated, only one tape unit is necessary. More than one 1402 may be simulated, each requiring one or two tapes, depending on the extent of simulation required. The input tape must contain card images in column binary format, 28 words to the card. The output tape will consist of similar card images containing the information intended for punching in the simulated output cards.

Modifier Bits: All Stacker Select Modifier bits are ignored and the Punch Feed Read feature is not simulated.

End-of-File Switch: The End-of-File switch is simulated as being always on, and the last card sequence specified for the 1402 is automatically put into effect whenever a tape mark is read on the input tape.

1442 Card Read Punch

Complete simulation of a 1442 Card Read Punch requires two tape units, one simulating card input, the other simulating card output. Both 729 tape units assigned to this 1442 must be on the same channel. If either input or output, but not both, is to be simulated, only one tape unit is necessary. More than one 1442 may be simulated, each requiring one or two tapes, depending on the extent of simulation required. The input tape must contain card images in column binary format, 28 words to the card. The output tape will consist of similar card images containing the original information from the input cards, together with any additional information intended for punching in the simulated output cards.

Modifier Bits: The Stacker Select Modifier bit is ignored, but the Eject bit, if present, causes simulated output of a card (if required) upon completion of punching.

End-of-File Key: The End-of-File key is not simulated, and the Last Card Sequence specified for the 1442 is automatically put into effect whenever a tape mark is read on the input tape.

1403 Printer

Output via the 1403 printer is simulated by writing the output on 729 tape in the form of BCD records. The FORTRAN character set is used, and the length of the print line is 132 characters. Characters in excess of 120 will be printed on a second line.

Carriage control codes are checked for validity and are simulated by inserting appropriate control characters in the print record. However, only one carriage control operation can be simulated for any one Write command; if more than one control code is encountered, only the last will be executed.

1443 Printer

Output via the 1443 Printer is simulated by writing the output on 729 tape in the form of BCD records. The FORTRAN character set is used, and the length of the print line may be either 120 or 144 characters, as specified by the programmer on an IOCF control card (see Appendix B). Characters in excess of 120 will be printed on a second line.

Carriage control codes are checked for validity and are simulated by inserting appropriate control characters in the print record. However, only one carriage control operation can be simulated for any one Write command; if more than one control code is encountered, only the last will be executed.

2311 Disk Storage Drive

One contiguous set of selected cylinders of a 2311 Disk Storage Drive may be simulated on selected cylinders of a 1301 Disk Storage (or 7320 Drum Storage) with cylinder mode option. (See description of IOCF card in Appendix.) The area of 1301 Disk Storage specified must have been previously formatted, using the 7090/7094 disk utilities, with one record per track; that record must have a data area of 465 words.

More than one 2311 drive may be simulated in a program, but activity on different simulated 2311's cannot be overlapped on the 7090.

The Simulator determines whether information it reads from the 1301 was written by the simulator, and simulates 2311 formats by 7090 programming. Each CCW chain causes the simulated track to be scanned from the index point.

Note: The Track Overflow option and the File Scan option are not simulated.

2401, 2402, 2403, and 2404 Magnetic Tape Units

Magnetic tape operations using 2401, 2402, 2403, and/or 2404 tape units are simulated using 729 tape units. The read backward operation is simulated subject to the following restrictions:

1. The number of bytes written must be a multiple of nine (even number of words in the input buffer), or the number of six-bit characters must be a multiple of six (integral number of words in the input buffer).

2. For seven-track operation, the data converter must be off.

Both standard nine-track and optional seven-track heads are simulated. All control and mode-modifier codes are checked and the appropriate actions initiated, except that densities will not be changed. Seven-track input tapes prepared by other means should normally contain an integral multiple of six 6-bit characters, to conform with usual 7090 practice.

Record Length: A 1,000-word buffer in the 7090 is used in tape simulation, permitting record sizes of up to 6,000 characters on 7-track tapes when not using the data converter feature, or 4,500 characters otherwise. An attempt to exceed buffer capacity will terminate the program. The Simulator does not attempt to deal with partial words; thus, if the last character in a record being written does not fall at the low-order end of a 7090 machine word, the record length will be extended (by filling the last partial word with zeros) to correspond to the end of the word. If such a record is subsequently read, it will appear to be of the correct length if the count is such as to cause Channel End at some point in the last 7090 word. Thus, a read count slightly different from the original write count for the same record will in some cases be accepted as correct.

File Protection: The 7090 provides no means for the program to determine whether a tape reel has a file protection ring; an attempt to write on a protected tape causes a 7090 machine delay. Thus, the file protection bit provided for actual System/360 tape operations cannot be simulated, and the user should insure that any tape on which he wants to write is not protected.

Simulation of Input/Output Interruptions

Termination of any input/output operation causes an interruption condition in the path to the input/output unit. This will cause an input/output interruption to be taken if the channel is enabled; otherwise, the interruption will remain stacked.

Channel Status Bits: The following channel status bits are simulated:

Program-Controlled Interruption
Incorrect Length
Program Check
Protection Check

Unit Status Bits: Unit status bits for the individual input/output units are simulated as specified in Appendix C for the various units.

Simulation of Program-Controlled Interruptions (PCI): Simulation of Program-Controlled interruptions is subject to restrictions similar to those that apply to the simulation of the Halt I-O instruction. It is not possible to give general rules for the effect of the PCI bit in any command, but attention is drawn to the following:

1. In a read operation with data chaining, a PCI bit in the first command will allow interruption before reading is completed. A PCI bit in any subsequent command in the chain will not be recognized until reading has been completed and the end of the data chain has been reached.

2. In a write operation with data chaining, a PCI bit in any command in the sequence will allow interruption before writing is completed. However, the end of the data chain will have been reached, and it will not be possible to append data to the current record.

SIMULATOR FACILITIES

The 7090 Simulator Program provides a number of features intended to facilitate the checking out of System/360 programs. These include the capacity to provide printouts of selected portions of main storage during execution of a program ("dynamic dump"), a facility for tracing the course taken by the program during execution ("flow trace"), protection against endless loops, and means for counting the number of times each type of instruction is obeyed and for punching this information for use in estimating System/360 running time for a given program.

DYNAMIC DUMP

The dynamic dump facility permits the programmer to obtain, whenever required, the contents of the following:

The current program status word
The storage protection keys (if the storage protect option is specified)
The general and floating point registers
One or more specified areas of main storage

This information is written on the System Output (SYSOU1). It can be presented in a number of different formats, as specified by the programmer. These include character, hexadecimal, packed and zoned decimal, and decimal floating-point notation; within certain limits, the information can be presented in units of halfwords, words, or double words, or, in some cases, in unit lengths specified by the programmer.

Each dump request includes a control number which is used in determining whether the dump is required during a particular running of the program. The control number is compared with the control numbers specified on DUMP control cards (see Appendix B). If a match is found, the dump request will be executed; otherwise, it will be ignored. In either case, simulation will resume with the instruction immediately following the dump request linkage. This procedure allows the programmer to assign various numbers to different dump requests and then to call for particular dumps by varying the numbers on the DUMP cards prior to loading.

To obtain a printout, the programmer must write a calling sequence that consists of a Supervisor Call instruction with identification code 1 (SVC 1), followed by a series of entries defining the dump requirements; this sequence specifies, among other things, the types of registers to be included in the dump. To obtain the contents of main storage, a Control List entry must define each area to be printed. (See the subsection "Control List.")

Calling Sequence

To enable execution of the dump routine, the program must place the following information in main storage in the format shown in Figure 1:

```
0          7 8   15 16       19 20   23 24    27 28        31
┌─────────────────────────────────────────────────────────────┐
|Supervisor |       | Call Seq |         |Reg     |  Control   |
|Call       |      1| Length   |        0|Option  |  No.       |
|-----------+-------------------------------------------------|
|           |                                                  |
|           |                                                  |
|Count      |       Address of Control List                    |
└─────────────────────────────────────────────────────────────┘
```

Figure 1.

This information is required for each dump that is to be executed. It is suggested that the programmer write a series of entries such as that given below. In this list, the small letters in the "Coding" column indicate variable information to be supplied according to the requirements of the particular dump. The appropriate coding is explained in the second column, under the heading "Variable."

CODING EXPLANATION

SVC 1

Purpose: to call the dump routine.

DC X'dd'

Purpose: to specify whether a Control List is present. (Actually, it indicates the length of the calling sequence.)

Variable: dd must be one of the following:
00 - No Control List is present.
C0 - A Control List is present.

DC X'rp'

Purpose: to specify (1) the registers to be written, and (2) a control number.

Variable: r must be one of the following:
0 - No registers.
1 - General registers only.
2 - Floating-point registers only.
3 - Both general and floating-point registers.

Variable: p must be a hexadecimal number from 0 to F. This number is compared with the numbers on the DUMP cards for the action deck. (For a discussion of action decks, see Section 3; for the format of the DUMP card, see Appendix B.) If a match is found, the dump is executed; otherwise it is ignored. By changing DUMP cards at load time, the programmer can select the dumps to be executed without having to reassemble the program to place new control numbers in the calling sequences.

DC AL1(ccc)

Purpose: to specify the number of Control List entries.

Variable: ccc must be a decimal number from 0 to 255 that gives the count of entries in the Control List that begins at the address in the following entry. If the dd code in the first entry after the SVC instruction above indicates that a Control List is present and if ccc in this entry is 0, all System/360 main storage specified by the CPUCF card (see Appendix B) will be written.

DC AL3 (aaaaaa)

Purpose: to specify the address of the first Control List entry.

Variable: aaaaaa is the address of the first entry in the list, normally written here as that entry's symbolic name.

NOTE: The last two entries are required if code dd in the first entry after the SVC instruction is C0; they should not appear otherwise. If these entries are not used, only the contents of the registers specified in the second entry following the SVC instruction will be written.

Control List

A Control List entry is required for each area of main storage to be written. It specifies the location and size of the area, together with the format of the dump. The necessary data must appear in main storage in the format shown in Figure 2:

```
 0  1 2  3 4     7 8     31 32    39  40              63
r--------------------------------------------------------┐
|   | |    |        |         |         |                |
|A  |B|   0| Format | First   | Length  | Count or       |
|   | |    |        | Address |         | Second Address |
|--------------------------------------------------------|
|                                                        |
|                                                        |
|          Header (optional)                             |
L--------------------------------------------------------┘
```

A = a code indicating whether a header follows

B = a code indicating whether bits 40-63 contain a count or an address

Figure 2.

The required information may be placed in storage by a series of entries such as that given below. The small letters in the "Coding" column indicate data to be supplied by the programmer; they are explained in the second column under the heading "Variable" for each entry.

CODING EXPLANATION

DC X'af'

<u>Purpose</u>: to specify whether a header follows the entry, whether the last field of the entry (the fourth item in this list) specifies a count or an address, and the dump format.

<u>Variable</u>: <u>a</u> must be one of the following:

0--No header follows the entry and the last field of the entry contains a count.
4--No header follows the entry and the last field of the entry contains an address.
8--A header follows the entry and the last field of the entry contains a count.
C--A header follows the entry and the last field of the entry contains an address.

<u>Variable</u>: <u>f</u> must be one of the following:

0--Hexadecimal; length must be specified in the second entry following this one.
1--Character (the low-order six bits are used to produce Character Set H); length must be specified in the second entry following this one.
2--Packed decimal; length must be specified in the second entry following this one.
3--Zoned decimal; length must be specified in the second entry following this one.
4--Hexadecimal halfwords with mnemonics.
5--Hexadecimal words without mnemonics.
6--Decimal floating-point, short format.
7--Decimal floating-point, long format.
8--Same as 4.
9--Halfword fixed-point decimal.
A--Full word fixed-point decimal.

DC AL3(aaaaaa)

<u>Purpose</u>: to specify the address of the first byte of main storage to be written.

<u>Variable</u>: <u>aaaaaa</u> is the address of the first byte to be written, normally expressed here as its symbolic label.

DC AL1(nnn)

<u>Purpose</u>: to specify the length of each item to be printed.

<u>Variable</u>: <u>nnn</u> must be a decimal number from 0 to 255; it gives, in bytes, the length of each item, if specified by any of the codes 0-3 in the second entry above.

NOTE: Although a length specification is not required for formats specified by codes 4-A, the space reserved for it must be accounted for. This may be done either by assigning an arbitrary length value (which will be ignored) or by changing the length of the following entry from AL3 to AL4.

DC AL3(bbbbbb)

<u>Purpose</u>: to specify either (1) the count of items to be written or (2) the address of the first byte following the last item to be written.

<u>Variable</u>: <u>bbbbbb</u> must be one of the following:

CODING EXPLANATION

1. A decimal number, if code a in in the first entry is either 0 or 8. In this case, the number will be interpreted as a count of items to be written ("item" means "half-word," "word," or "double-word," depending on format code specified in second entry, assuming it is any of the codes from 4 through A). In other cases, the length specification in the preceding entry is controlling.

2. A symbolic label designating the address of the first byte following the last item to be written, if code a in the first entry is either 4 or C.

DC C'cccccccc'

Purpose: to specify a header to be written at the beginning of the dump. The header need not be the same as the label of the first byte of the dump.

Variable: cccccccc may be any combination of 8 characters that can be printed by the 716 Printer, including the blank.

NOTE: This header is not required and may appear only if either code 8 or C is used in the first entry.

The dynamic dump also will show the calling sequence that initiated the dump, the current program status word, and each Control List entry as it is processed. The order of a typical dump would be as follows:

Calling sequence and program status word
General registers
Floating-point registers
First Control List entry
First main storage dump
Second Control List entry
Second main storage dump if specified
.
.
.
Last Control List entry
Last main storage dump

FLOW TRACE FACILITY

The flow trace facility permits the programmer to trace the execution of a program at object time in specified areas of main storage. This facility provides for recording details that relate to interruptions, branches, changes of mode, and certain conditions that may result in changes of flow. To obtain this information, the programmer must define one or more "flow trace areas" by specifying the first and last addresses of each area on a FLOW card. (The format of a FLOW card is given in Appendix B.) Thereafter, flow trace information will be recorded if any of the following types of activities should occur within a flow trace area:

1. A successful branch into, within, or out of the area.

2. A Supervisor Call instruction.

3. An Input/Output interruption.

4. A Program interruption.

The information recorded will consist of the relevant items from the following list, in the order stated:

1. The address of the last instruction obeyed, if number 3 in this list is given; otherwise, the address stored in the old program status word.

2. In the case of an interrupt, the stored old program status word; or, in the case of a branch, the program status word at the start of the operation.

3. Where possible, the last instruction obeyed, in hexadecimal with mnemonic; or, for input/output interruptions, the channel status word.

4. The address of the next instruction to be obeyed.

5. The new program status word.

6. The cause of the interruption.

Items 1 and 4 are given, where possible, in the form of segment name plus four-character hexadecimal offset. Where this is not possible, the items are given as an absolute address of six hexadecimal characters.

NORMAL PROGRAM TERMINATION

Normal termination of a program is specified by the use of a Supervisor Call instruction with identification code 0 (SVC 0). This instruction will be considered by the Support Package not as a normal SVC instruction but as a special request to the Support Package to terminate simulation of the current program and to load and commence simulation of the next program. (Further information is given in Section 4.)

INFINITE LOOP PROTECTION

The infinite loop protection facility is designed to prevent a program from being caught in an endless loop of System/360 machine instructions. The programmer may specify the maximum number of instructions to be executed, using an ICNT card (see Appendix B.)

The number specified refers to the number of actual operations performed, not to the number of instructions in the object program listing. If this count is exceeded, the flow trace procedure described above is initiated over all of the simulated main storage specified on the CPUCF card (see Appendix B), and after 50 flow trace events in the System/360 program have been traced and recorded, execution is ended by writing a hexadecimal dump of all main storage that has been specified. The next program is then loaded.

If the ICNT card is not provided, the Simulator will assume a count that corresponds approximately to five minutes of 7090 operating time.

INSTRUCTION COUNT

If the programmer so specifies, the Simulator will maintain a count of the number of times each type of System/360 machine instruction is executed. This facility may be called in by specifying the DETAIL option on an ICNT control card, as shown in Appendix B. When a count is called for, a table is set up that contains an entry for each possible type of machine instruction. Each attempted execution, whether interrupted or successfully completed, is counted, and at the end of program execution, a list is printed showing the count of all instructions except those with a count of zero. The printed list also shows the corresponding mnemonic and hexadecimal codes for each instruction.

When an Execute instruction is simulated, the count is incremented for both the Execute instruction and its subject instruction.

TIMING INFORMATION

An instruction count obtained as described above can be punched directly into cards by the on-line 721 Card Punch. Columns 1-70 of each card are divided into five fields, each having 14 columns. Each of the five fields is subdivided to contain the hexadecimal operation code in the first two columns; the instruction count, in decimal and right-justified, in the next 11 columns; and the last column blank. Columns 71-72 contain a decimal serial number and columns 73-80 are blank.

Each group of cards so punched is preceded by a card with the words "COUNT OF INDIVIDUAL INSTRUCTIONS." punched in columns 2-34. Other columns of this card are blank.

The card deck is obtained by specifying the TIME option on an ICNT control card, as explained in Appendix B.

OPERATOR FACILITIES

Facilities are provided to permit the operator both to terminate the execution of a program and to interrupt it.

TERMINATION

The execution of a program may be terminated by setting Sense Switches 5 and 6 on. A message indicating abnormal termination is written on the System Output (SYSOU1), together with a hexadecimal dump of simulated main storage. The same message is printed on line, and simulation of this program, and all programs remaining in the current "Action" (see Section 3) is discontinued.

INTERRUPTIONS

Interruptions may be simulated by setting 7090 Sense Switches on as follows:

7090 SENSE SWITCH	RESULTING INTERRUPTION (SIMULATED)
2	1052 Printer-Keyboard Attention Signal
3	IBM System/360 Console Interruption
4	External Interruption

Sense Switch 6 must be set on to activate any of these interruptions.

Each interruption causes an on-line message to be printed requesting the operator to reset the Sense Switches. The 7090 then pauses and may be restarted by pressing the Start key.

MESSAGES

Messages are printed either on line, using the 716 Printer, or off line, on the System Output (SYSOU1). On-line messages are used both to simulate the Console Printer and to inform the operator of various conditions that may arise during simulation. Off-line messages are used to furnish information of interest to the programmer; such messages are used to signal error conditions, the start of program execution, and indications of any actions taken by the operator. (See Appendixes E and G.)

SECTION 3: DECK STRUCTURE FOR INPUT TO THE 7090 SUPPORT PACKAGE

Although the 7090 Assembly Program and the 7090 Simulator Program are separate and independent programs and have been discussed as such, they are in fact one system and may be used either independently or in conjunction with each other, at the discretion of the programmer. The method for doing so is explained later in this section, under "ACTION Cards." In all cases, the two programs are controlled by a supervisory program operating under the 7090 System Monitor.

This section deals primarily with the structure of the card deck which, when written on tape by the 1401 Input Program, becomes input to the 7090 Assembly and Simulator Programs. The discussion is concerned with the use of the control cards required by the 7090 System Monitor and the Support Package, and with the organization of the input deck. The actual formats of the control cards are given in Appendix B, together with a reference summary of their functions.

These various elements are discussed in greater detail in the remainder of this section.

GENERAL CONSIDERATIONS AFFECTING INPUT

Input to the Assembly Program is handled directly by that program. Programs to be simulated are placed in main storage by a loader provided for that purpose. All input must be prepared originally in card form for conversion to tape by the 1401 Input Program. Therefore, this section deals with the organization of the input in its original card form.

RELOCATION

As explained in Section 1, under "Output from the Assembly Program," the card deck resulting from the use of that program is in relocatable format, with facilities provided as required to permit the loader to assign absolute addresses at load time, as well as to permit cross references to be made between two or more program segments that are loaded together. The term "segment" has been defined as the input to, or the output from, one execution of the 7090 Assembly Program, and a "program" consists of one or more segments loaded together into simulated System/360 main storage for execution.

The loader normally will begin loading each program from simulated System/360 location 100 to the base 16. This can be changed by use of a BASE card (see Section 4).

Upon loading, all segments will be assigned absolute addresses. For those that require relocation, the loader uses a Control Dictionary derived for each segment from the External Symbol Dictionary cards, which define the external and internal symbols referred to in the segment, and the Relocation List Dictionary cards, which define the locations of addresses requiring relocation (see Section 1).

ENTRY POINT

After loading is completed, control will transfer to the program at an entry point determined by one of the following, given in the order of priority in which the loader will accept them:

1. The address, if any, on the Load Terminate card (see the discussion of that card later in this section).

2. The first address, if any, encountered on a Load End card. If the deck contains more than one Load End card and more than one address is given, all addresses but the first will be ignored.

3. The address of the first byte of the program as loaded.

DECK STRUCTURE

Input decks for use by the Support Package must be introduced and followed by the control cards required for any program running under the 7090 System Monitor, as explained in the publication *IBM 7090/7094 IBSYS Operating System, System Monitor (IBSYS)*. (An abbreviated summary of the relevant cards is given in Appendix B.) They also must include control cards relating particularly to the operations of the Support Package, as explained in this section. This is the basic structure of the input deck:

1. 7090 System Monitor control cards

2. $EXECUTE SUPPAK card

3. One or more "action decks," as defined later in this section

4. $IBSYS card or other 7090 System Monitor control card that returns control to the 7090 System Monitor

7090 SYSTEM MONITOR AND $EXECUTE SUPPAK CARDS

The first cards in the input deck must be the 7090 System Monitor control cards specified for the operation of that system. These must be followed by a $EXECUTE SUPPAK card, which calls in the Support Package supervisor. The supervisor in turn calls in the Assembly and Simulator programs, as required. The supervisor can return control to the 7090 System Monitor when required, and, as a means of maintaining communication with that monitor, it recognizes and acts correctly on the following 7090 System Monitor cards:

7090 SYSTEM MONITOR CONTROL CARD	ACTION
$IBSYS $STOP $JOB $EXECUTE (other than $EXECUTE SUPPAK)	Terminates present action; returns control to 7090 System Monitor.
$EXECUTE SUPPAK	Terminates present action and begins the next.
$ID	Transfers control to an installation accounting routine; does not terminate present action.

In addition, the supervisor recognizes the following 7090 System Monitor cards but retains control:

7090 SYSTEM MONITOR CONTROL CARD	ACTION
$*	Causes writing of on-line messages.
$PAUSE	Results in a pause by the 7090.

ACTION Cards

The Support Package system can operate in three modes: assembly only, simulation only, and successive assembly and simulation. To specify which mode is to be employed, the programmer must place an ACTION card preceding each program or group of programs to which it applies. This card may have one of the following three forms:

```
S   ACTION ASSEMBLE
S   ACTION SIMULATE
S   ACTION ASSEMBLE,SIMULATE
```

The ACTION card must follow immediately after either a $EXECUTE SUPPAK card, or the last card of the preceding action deck. The "action" it specifies (i.e., the assembly, simulation, or successive assembly and simulation) is terminated by any of the following: (1) an ACTION card for the next action, (2) any of the 7090 System Monitor control cards that return control to the 7090 System Monitor, or (3) abnormal termination of simulation by the operator, the simulated program, or the Simulator.

The specific effect of each of the three types of ACTION cards is described in the following text.

The Assemble Action

When assembly only is specified, all input cards except those that make up program segments in symbolic source language are ignored. Thus, it is possible to submit a deck that contains both unassembled segments and other components of the object program, and only those segments requiring assembly will be acted upon. As noted in Section 1, a listing of the assembled segments will be written on the System Output (SYSOU1), and the assembled object program will be written in card-image format on System Punch (SYSPP1). These tapes will not contain the material that had been assembled previously. The SYSPP1 tape may then be processed on a 1401 to prepare a card deck for subsequent input to the Simulator Program. The tape itself may not be used directly as input. In this respect, the ASSEMBLE action differs from the ASSEMBLE,SIMULATE mode, which prepares a tape (SYSUT2) that is used immediately for input to the Simulator.

The structure of an action deck for use in the ASSEMBLE mode may be represented as follows:

1. ACTION ASSEMBLE card

2. Any material except symbolic segments

3. One or more segment decks, each consisting of:

    a. TITLE card (see Section 1)

    b. Symbolic cards to be assembled

    c. END statement card (see Basic Assembler Language publication)

d. End-of-File card (see Appendix B)

e. Any material except symbolic segments

The Simulate Action

When simulation only is specified, all segments composing the program or programs to be simulated must be in assembled form on the System Input (SYSIN1). The loader will then place the assembled segments in simulated System/360 main storage. The Load Terminate card indicates to the loader that the last segment has been loaded and that simulation is to begin. If serious errors are discovered during loading, no further simulation will occur during the course of the action, but loading will continue until the end of the action deck is reached.

The structure of an action deck for use in this mode may be represented as follows:

1. ACTION SIMULATE card

2. One or more program decks, each consisting of:

   a. S control cards applicable to the program*

   b. One or more segments, each consisting of:

      S control cards applicable to the individual segment*

      Assembled segment

   c. Load Terminate card (see "Load Terminate Card," later in this section)

   d. Data (optional)

   e. End-of-File card (see Appendix B)

The Assemble, Simulate Action

When successive assembly and simulation are desired, the input deck should contain at least one segment in unassembled form and may also include any number of assembled program decks. While the Support Package will accept decks in which there is no material requiring assembly, the SIMULATE mode would be preferable in such a case.

Processing in the ASSEMBLE,SIMULATE mode requires two passes, whereas each of the other two modes requires only one. The first pass is similar to that used in the ASSEMBLE mode, except that, instead of ignoring those portions of the deck that do not require assembly, the Assembly Program will copy them onto System Utility Unit 2 (SYSUT2), while those portions specified for assembly will be written as assembled decks on both SYSUT2 and SYSPP1. The first pass continues until another ACTION card (or a 7090 System Monitor control card that would normally return control to the 7090 System Monitor) has been copied onto SYSUT2. The SYSUT2 tape is then rewound and the second pass begins. The second pass is identical to that performed in the SIMULATE mode, except that SYSUT2 is used as the System Input file, instead of SYSIN1.

*See "Other Support Package Control Cards," later in this section.

Certain serious program errors, if discovered during the first pass, will cause omission of the second pass. All remaining symbolic segments will be assembled. (See Appendix G.)

The structure of an action deck for use in the ASSEMBLE,SIMULATE mode is as follows:

1. ACTION ASSEMBLE,SIMULATE card

2. One or more program decks, each consisting of:

   a. S control cards applicable to the program*

   b. One or more segments, in any order, each consisting of:

      S control cards applicable to the individual segment*

      An assembled or symbolic segment as shown in the formats for ACTION ASSEMBLE and ACTION SIMULATE

   c. Load Terminate card (see "Load Terminate Card," later in this section)

   d. Data (optional)

   e. End-of-File card (see Appendix B)

OTHER SUPPORT PACKAGE CONTROL CARDS

Control cards relating particularly to the 7090 Support Package may be grouped into two general categories: "System Control cards" and "Simulator Control cards," depending on their basic functions. All of these cards require the character S in column 1 and are therefore referred to in this manual as "S control cards" or "S cards." There are four System Control cards, as follows:

ACTION card
COUNT card
OUTPUT card
TITLE card

The ACTION cards have been described earlier in this section, the COUNT, OUTPUT, and TITLE cards are described in Section 1, and the actual formats of the cards are given in Appendix B.

The following text contains a brief description of the Simulator control cards.

BASE Card

The BASE card may be used in connection with supervisor programs and other special programs written in System/360 coding, as explained in Section 4. In general, it is used when such a supervisor program is present to specify an address at which programs other than the supervisor are to be loaded.

*See "Other Support Package Control Cards" on this page.

CPUCF Card

The CPUCF card is used to specify the configuration of the central processing unit (CPU) of the System/360 for which the object program is intended. Details are given in Appendix B.

DUMP Card

The DUMP card is used to specify which dynamic dumps in an object program are to be performed. Each dump request contains a control number, which is matched against control numbers in the DUMP cards for the program. If a match is found, the dump is executed; otherwise, the dump request is ignored. See Section 2 for details.

FLOW Card

FLOW cards may be used to define areas of main storage in which flow traces are requested, as explained in Section 2. There may be as many as ten such cards for each program segment. Further details are in Appendix B.

ICNT Card

The ICNT card is used in connection with the infinite loop protection feature described in Section 2. It specifies a maximum number of instructions to be executed before the program is terminated. It may also be used to specify that a count be made of individual System/360 machine instructions executed. Details are given in Appendix B.

IOCF Card

The IOCF card is used to relate specific 7090 machine components to specific System/360 input/output devices, as explained in Appendix B.

INITL Card

The INITL card is used to specify the leftmost 40 bits of the initial PSW, as explained in Appendix B.

IPL Card

The IPL card is used to specify simulation of the Initial Program Loading procedure. When encountered, it results in immediate action. See Section 4 for further details.

SUP Card

The SUP card is used to specify the effect of certain Supervisor Call instructions when the user is employing supervisor programs of his own in System/360 coding. This subject is discussed under "Advanced Considerations," Section 4.

POSITION OF CONTROL CARDS

In general, those S control cards that apply to an entire program may be placed at the beginning of the program, while those that apply only to a particular segment should be placed at the beginning of the segment. Except in the case of the IPL card, which is acted upon immediately after it is encountered, the order in which the control cards occur is immaterial. However, the following general rules should be noted:

1. A control card will apply only to the program or segment in which it occurs, except as specified in 2 and 3 below. If there is more than one control card of the same type, their effects will be "cumulative"; that is, all provisions specified remain in effect until the end of the program or segment, except that in case of conflict, the last specification given is governing.

2. An IOCF card applies to all programs and segments in the same action.

3. If a control card appears in a supervisor program written in System/360 coding, as explained in Section 4, its effects will apply to all subsequent programs and segments.

LOADER CARDS

In addition to the preceding cards, there are several cards that pertain directly to the operations of the loader. These are described in the following text.

Load Terminate Card (LDT)

Any program that will require simulation, whether in assembled or unassembled form, must contain a Load Terminate (LDT) card immediately following the last segment to be loaded. This card must be supplied by the programmer; it is not provided as output from the Assembly Program. The format of the LDT card is specified in Appendix B. The function of this card is to indicate to the loader that simulation is to begin. The Load Terminate card may, at the programmer's option, also specify the entry point to the program.

Optional Cards

Additional facilities may be obtained by the use of either or both of the cards described in the following text.

REPLACE CARD (REP): With the Replace card, the programmer can substitute new text for portions of assembled text. Each Replace card must contain the assembled address of the first byte to be replaced and may contain from 1 to 22 bytes of text. This text will be substituted, byte for byte, for the original text, beginning at the address specified. Both the new address and the new text must be stated in hexadecimal, using the format given for the card in Appendix B. Replace cards should follow all of the text cards in the segment, but must appear before the Load End card for the segment. Addresses will be relocated as for the other text cards of the segment.

INCLUDE SEGMENT CARD (ICS): The Include Segment card may be used to reserve storage space for a segment to be loaded subsequently. The ICS card must be prepared in the format specified in Appendix B, giving the name and length of the segment. ICS cards are normally placed after the S control cards for the first segment of the program for which the ICS cards are to have effect. However, they will be recognized at any point within an assembled segment.

SECTION 4: ADVANCED CONSIDERATIONS

In the previous discussion, it has been assumed that the programs being simulated are relatively simple in nature and that they are brought into simulated System/360 storage by the loader in the 7090 Support Package. It has also been assumed that each program in an action is independent of other programs in the same action and that each will be terminated normally, as explained in Section 2, by the use of an SVC 0 instruction. Such an instruction not only terminates simulation but also returns control to the Support Package supervisor in order to load the next program.

However, the Support Package is designed in such a way that the programmer may use special programs in System/360 machine language, including programs that perform functions normally associated with supervisor and control programs. Such programs may be used to control other programs. For example, the programmer may wish to employ a special supervisor that contains a loader, in lieu of the supervisor and loader facilities of the 7090 Support Package. This section describes how programs of this kind can be loaded, called in, and simulated, as required. It further describes an alternative method of loading object programs, using a simulation of the Initial Program Loading procedure that is provided on an actual System/360.

SIMULATION OF SUPERVISOR CALL INSTRUCTIONS

To understand the concepts of this section, it is necessary to understand that the 7090 Simulator Program provides two methods of handling Supervisor Call instructions having the identification codes 0, 1, 2, and 3.

In one method, which simulates the manner in which System/360 actually executes an SVC instruction, the old program status word is stored and a new one is obtained, and the program continues as specified by the new PSW. In this case, interpretation of the Supervisor Call is carried out by a series of instructions specified in System/360 coding, and it is the responsibility of the user to provide whatever program facilities are required in handling the interruption. When an SVC instruction is simulated in this manner, the Simulator may be thought of as operating in "System/360 mode."

In the second method, a Supervisor Call instruction is executed by calling in a specific 7090 routine provided by the Simulator for the purpose. In this case, the Simulator may be considered to be operating in "7090 mode."

These two methods are applicable to Supervisor Call instructions with the identification codes 0, 1, 2, and 3, each of which can operate in either 7090 or System/360 mode. A standard mode is set for each of these when the Simulator Program is called in, but the mode can be changed by the use of SUP cards, as explained in the following text. As initially set, SVC 0 and SVC 1 operate in 7090 mode and SVC 2 and SVC 3 operate in System/360 mode. These modes are set whenever an ACTION card is encountered.

The general effects of these instructions, when operating in 7090 mode, are summarized in the following table:

INSTRUCTION	EFFECT IN 7090 MODE
SVC 0	Terminates simulation of the current program, spaces System Input to next program, and loads that program.

INSTRUCTION	EFFECT IN 7090 MODE
SVC 1	Calls for a dynamic dump (see Section 2) and then continues simulation of the present program.
SVC 2	Terminates simulation of the present program and spaces the System Input to the beginning of the next action.
SVC 3	Has varying effects, depending on the use of BASE and SUP cards, as explained later in this section. A primary purpose is to permit control card processing, followed by a return to simulation.

SUP Card

The basic function of a SUP card is to change the mode in which a subsequent SVC instruction will operate. It has the following format:

```
S      SUP       variable field
```

The variable field may contain from 1 to 4 subfields, each consisting of one of the SVC identification codes 0, 1, 2, or 3. Wherever a code is written as an unsigned integer, it specifies 7090 mode; wherever it is preceded by a minus sign, it specifies System/360 mode. A code specified in a SUP card overrides any previous mode specification for the same instruction but does not change the mode of any other. For example, the card S SUP -0,3 would establish System/360 mode for SVC 0 and 7090 mode for SVC 3, but would not change whatever modes had previously been in effect for SVC 1 and SVC 2. Once set, either by an ACTION card or a SUP card, the mode remains in effect until altered by a subsequent ACTION or SUP card. Whenever an SVC instruction is encountered in the object program, it will be executed in whatever mode has been set for it. Such instructions, it should be noted, are written in the source program with positive identification codes, regardless of the mode in which they will operate.

The SUP card format is given in Appendix B.

BASE Card

Unless otherwise directed, the loader in the 7090 Support Package will place the first byte of an object program at simulated System/360 location 100 to the base 16 (i.e., decimal 256), loading the program from that point. This normal loading assignment can be overridden by the use of a BASE card, which has the following format:

```
S      BASE      variable field
```

The variable field must specify an absolute System/360 address in hexadecimal. This address becomes effective for the program following the one in which it occurs and remains in effect until another BASE card is encountered or until the end of the action deck is reached.

Uses of SUP and BASE Cards

If the programmer wishes to use a supervisor program in System/360 coding, he may wish any SVC 0 instruction in a subsequent object program to call on this supervisor, rather than the facilities in the 7090 Support Package supervisor. This requires that the SVC 0 instruction in the object program be able to operate in System/360 mode. In these circumstances, the supervisor may still need to call for the processing of S control cards, and also (if it does not contain a loader) for the loading of other programs. For this to be possible, it is necessary that there be available an SVC instruction that will be interpreted in 7090 mode and will have the same effect as an SVC 0 instruction operating in that mode. Since the SVC 0 instruction has already been set to operate in System/360 mode, it is necessary to use another instruction, SVC 3; to set the proper modes for both instructions, the following coding should be used:

```
S       SUP      -0,3
```

A supervisor of the kind described may itself be loaded either by an Initial Program Loading procedure (see the end of this section) or by the Support Package.

A primary function of the BASE card is to prevent the loading of a program into an area that is still in use. For example, a BASE card could be employed to protect a supervisor from inadvertent obliteration. However, if such a supervisor contains a loader, it will presumably contain provisions for protecting itself. In such a case, the use of a BASE card would be unnecessary.

It should be noted that BASE and SUP cards have interrelated effects. Their presence together, or the absence of either or both, has specific meanings to the supervisor, particularly in determining the effect of any SVC 3 instruction. The effect of that instruction, when operating in 7090 mode, is summarized in the following table:

CONTROL CARDS ENCOUNTERED	ASSUMPTIONS MADE BY 7090 SUPPORT PACKAGE SUPERVISOR	EFFECT OF SVC 3 WHEN OBEYED IN 7090 MODE
SUP -0 alone	The program is a supervisor containing a loader.	The Control Card Processor in the Support Package is used to process the control cards, and loading and simulation are resumed by returning control to the System/360 supervisor.
BASE alone or both BASE and SUP -0	The program is a supervisor not containing a loader.	The Control Card Processor in the Support Package is used to process the control cards, and the Support Package loader then loads the next program, starting at the address specified in the BASE card, and enters it.
Neither BASE nor SUP -0	The program is not a supervisor.	Control is returned to the Support Package for both control card processing and loading of the next program, which will begin at standard location 100 to the base 16 in simulated storage.

INITIAL PROGRAM LOADING (IPL)

Simulation of the Initial Program Loading procedure (IPL) available in the System/360 may be initiated by using an IPL control card (see Appendix B). This card terminates the reading of control cards by the Support Package and causes execution to begin with an IPL operation from the input/output unit specified on the control card. This input/output unit may be any 7090 System Monitor unit, including SYSIN1.

The IPL light on the console will be simulated by 7090 Sense Light 1, which will remain on during the IPL procedure. If an error is detected during the course of the procedure, an error message will be written on the System Output (SYSOU1), and execution of subsequent programs within that action will be inhibited. Loading will continue, however, until the next ACTION card (or 7090 System Monitor control card returning control to that monitor) is read. An exception to this is that if a supervisor program in System/360 coding has been loaded in accordance with the procedures discussed earlier in this chapter, its loading functions will cease to be performed after an error occurs.

The program used in IPL procedure should have the following structure:

1. S control cards for the program, of which the last must be the IPL card (see Appendix B).

2. Binary deck to be loaded by the IPL procedure. This deck may be on any 7090 unit assigned to a specified System/360 unit by an IOCF card (see Appendix B).

3. Data (optional).

4. End-of-File card.

SECTION 5: 1401 INPUT PROGRAM

The 1401 Input Program is used to read input card decks in the form described in Section 3 and prepare an input tape for use by the 7090 Support Package. The tape will consist of card-image records; some of these will be in binary coded decimal (BCD) representation, others in binary mode. (For information on 1401 operation, see Appendix F.)

BCD card images may be either blocked or unblocked. In general, control cards are written on tape in unblocked form, while Basic Assembler Language source cards are written in blocked form. Each card image is written on tape as 84 characters. The last 4 characters of each record, whether blocked or unblocked, contain "look-ahead bits," which indicate the mode of the record or block that follows.

Each binary tape record contains 168 characters and will occupy 28 IBM 7090 machine words; the look-ahead bits are placed in the last 8 characters.

The structure of the input deck must be as described in Section 3. Cards that the 1401 Input Program finds between a $EXECUTE SUPPAK card and either a $IBSYS card or another 7090 System Monitor control card that causes a return of control to the 7090 System Monitor will be treated as cards intended for processing by the Support Package programs.

The 1401 Input Program will act upon the cards as follows:

Cards for Processing by the Support Package Programs

TYPE OF CARD	WRITTEN AS
All cards with 12-2-9 punches in column 1, including Load Terminate card (these punches identify loader cards)	Binary record
All cards following a Load Terminate card, up to, but not including, the next End-of-File card	Binary record
All cards following an IPL card, up to, but not including, the next End-of-File card	Binary record
All cards following a TITLE card, up to, but not including, the next End-of-File card	Blocked BCD record
End-of-File card (must have EOF in columns 2-4 in addition to 7 and 8 punches in column 1)	Tape mark
All other cards, including TITLE and IPL cards	Unblocked BCD record

Cards Not Intended for Processing by the Support Package Programs

TYPE OF CARD	WRITTEN AS
Binary cards (identified by 7 and 9 punches in column 1)	Binary record
End-of-File card (identified by 7 and 8 punches in column 1); the symbol EOF may occur in columns 2-4, but it is not required	Tape mark
All other cards	Unblocked BCD record

APPENDIX A: USE OF 7090 I/O UNITS BY 7090 ASSEMBLY AND SIMULATOR PROGRAMS

The following chart shows the 7090 input/output units used by the 7090 Support Package, giving the names assigned those units when used by the 7090 System Monitor, together with the functions they serve in Support Package operations.

The programs of the Support Package issue no density-setting instructions to 729 tape units. After each assembly, SYSUT1 and SYSUT3 are rewound; after an assemble and simulate action, SYSUT2 is rewound.

System/360 input/output devices may be simulated on any 7090 System Monitor tape units on Channel A and/or B. The use of SYSIN1, SYSOU1, and SYSPP1 is recommended for simulation of 1402 or 1442 input, 1403 or 1443 output, and 1402 or 1442 output, respectively, because of the close relationship between their uses in simulation and their normal uses as 7090 System Monitor units. This usage is not mandatory, however.

Checking of 7090 System Monitor names on IOCF control cards (see Appendix B) proceeds as follows:

1. SYSPCH, SYSCRD, SYSPRT and SYSLB1 are rejected.

2. SYSUT2 is rejected in an assemble and simulate action, since during simulation, SYSIN1 is interpreted to mean SYSUT2.

3. The unit is rejected if it is not a 729 tape unit on Channel A or B, or a 1301 Disk Storage (7320 Drum Storage) on any channel.

TABLE 1 (PART 1). USES OF INPUT/OUTPUT UNITS

NAME AND FUNCTION OF UNIT	SYSCRD (SYSTEM CARD READER)	SYSPRT (SYSTEM PRINTER)	SYSLB1 (SYSTEM LIBRARY UNIT 1)	SYSIN1 (SYSTEM INPUT UNIT 1)
Type of unit	711	716	729 or other device (see introduction)	729
Always used for	Simulation of 1052 input	Simulation of 1052 output; on-line messages	IBSYS system containing Assembly and Simulator Programs	Control cards and program deck input
Always may be used for	--	--	--	Simulation of 1402 or 1442 input
Assembly Program uses for	--	--	--	--

NAME AND FUNCTION OF UNIT	SYSCRD (SYSTEM CARD READER)	SYSPRT (SYSTEM PRINTER)	SYSLB1 (SYSTEM LIBRARY UNIT 1)	SYSIN1 (SYSTEM INPUT UNIT 1)
Assemble-Simulate action uses for	--	--	--	Source for rewriting on SYSUT2
May conditionally be used for	--	--	--	--

TABLE 1 (PART 2)

NAME AND FUNCTION OF UNIT	SYSOU1 (SYSTEM OUTPUT UNIT 1)	SYSPP1 (SYSTEM PERIPHERAL PUNCH UNIT 1)	SYSUT1 AND SYSUT3 (SYSTEM UTILITY UNITS 1 AND 3)	SYSUT2 (SYSTEM UTILITY UNIT 2)
Type of Unit	729	729	729	729
Always used for	Dump, trace, and instruction count information	--	--	--
Always may be used for	Simulation of 1403 or 1443 output	Simulation of 1402 or 1442 output	--	--
Assembly Program uses for	Assembly listing	Assembled deck	Work tape	--
Assemble-Simulate action uses for	--	--	--	Rewritten SYSIN1 file
May conditionally be used for	--	--	Any use	Any use, if action is not assemble-simulate

APPENDIX B: CONTROL AND LOADER CARDS

This appendix summarizes the format requirements and operational uses of the control and loader cards specified for the 7090 Support Package. The cards are listed in the following order:

7090 System Monitor control cards
Support Package control cards (S cards)
End-of-File card
Loader cards

Within each group, the cards are arranged in alphabetical order.

FORMAT CONVENTIONS

The following conventions should be understood in using the basic formats in this appendix:

1. A number appearing above an entry in a format specifies the number of the card column in which that entry must begin.

2. Entries printed entirely in capital letters indicate actual coding. Where such coding is indicated, it must be punched exactly as shown.

3. Entries printed in small letters are either dummy entries or explanations of coding that is required; they indicate fields for which the programmer must supply the actual coding. In some cases, the acceptable coding is limited to certain specific choices, which are given below the format. In other cases, the nature of the coding is variable, but the rules are given below the format. In either case, the requirements are shown under the heading Subfields, wherever required.

4. In certain cases, coding and/or dummy entries are enclosed in square brackets, i.e., []. This notation indicates that the field may be used or omitted as required by the particular circumstances. In certain other cases, two or more entries are placed one above the other and enclosed in curved braces, i.e.,{ } ; this notation indicates that the programmer should use one, but not more than one, of the enclosed entries.

5. Although punctuation is not shown in every format, the following rules must be observed:

   a. A field is terminated by the first blank encountered; if a blank occurs after column 15 on a 7090 System Monitor (IBSYS) card or after column 21 on an S card, no subsequent field on the card will be recognized.

   b. If a variable field on an S card (i.e., a field beginning in column 22) contains more than one subfield, the subfields must be separated by commas; no blanks are permitted between them.

6. No entry may extend to the right of column 71.

CONTROL CARDS

7090 SYSTEM MONITOR CONTROL CARDS

The 7090 System Monitor control cards are explained in detail in the reference manual IBM 7090/7094 IBSYS Operating System, System Monitor (IBSYS). The listing below covers those that are recognized by the Support Package and is limited to a brief indication of their use, for reference purposes, and to a statement of their formats.

$EXECUTE Card

Purpose: To call in a specified system.

Format:

```
1                   16
$EXECUTE            system name
```

Subfield: The system name may be either SUPPAK (the name assigned to the 7090 Support Package supervisor for this purpose) or the name of a system acceptable to the 7090 System Monitor; see the manual on that system.

$IBSYS Card

Purpose: To call in the 7090 System Monitor.

Format:

```
1
$IBSYS
```

$ID Card

Purpose: To specify a transfer of control to the installation accounting routine under conditions specified in the 7090 System Monitor reference manual.

Format:

```
1                   16
$ID                 variable field (see 7090 System Monitor
                    reference manual)
```

$JOB Card

Purpose: To reinitialize certain unit addresses in the 7090 System Monitor, test for operator interruptions, and transfer control to the installation accounting routine.

Format:

```
1                   16
$JOB                variable field (see 7090 System Monitor
                    reference manual)
```

$PAUSE Card

Purpose: To cause the 7090 to print an on-line message to the operator and then pause.

Format:

```
1                   16
$PAUSE              message
```

Subfield: The message may consist of any desired instructions to be given to the operator.

Note: As executed under the 7090 Support Package, the 7090 will pause, but control will be retained by the Support Package supervisor.

$STOP Card

Purpose: To cause an end-of-jobs sequence to occur.

Format:

```
1
$STOP
```

$* Card

Purpose: To permit the programmer to enter comments on-line.

Format:

```
1                   3 (or any column thereafter)
$*                  comments
```

Subfield: The comments may consist of any message the programmer wishes to have printed on-line.

Note: As executed under the Support Package, control remains with the Support Package supervisor.

7090 SUPPORT PACKAGE CONTROL CARDS (S CARDS)

For convenience, the general rules governing S cards are repeated below:

1. A control card will apply only to the program or segment in which it occurs, except as specified in Rules 2 and 3. If there is more than one

control card of the same type, their effects will be cumulative; that is, all provisions specified remain in effect until the end of the program or segment, except that in the case of conflict, the last specification is governing.

2. An IOCF card applies to all programs and segments in the same action.

3. If a control card appears in a supervisor program written in System/360 coding, as explained in Section 4, its effects will apply to all subsequent programs and segments.

ACTION Card

Purpose: To specify whether an input deck is to be processed in the assemble only, simulate only, or assemble-simulate mode.

Format:

```
1          14       / 22
                   |  ASSEMBLE
S          ACTION <   SIMULATE
                   |  ASSEMBLE,SIMULATE
```

Subfields: The coding is self-explanatory, but further details will be found in Section 3. If any subfield other than ASSEMBLE or SIMULATE is specified, the ASSEMBLE,SIMULATE mode will be set.

BASE Card

Purpose: To specify an address at which an object program is to be loaded (see Section 4).

Format:

```
1          14         22
S          BASE       hhhh
```

Subfield: hhhh = a hexadecimal value of from 1 to 4 digits, specifying an address in simulated System/360 main storage. Relocatable programs after the present one will be loaded beginning at the address specified.

COUNT Card

Purpose: To specify the number of cards in a source deck (see Section 1).

Format:

```
1          14         22
S          COUNT      decimal integer
```

CPUCF Card

Purpose: To specify the configuration of the System/360 CPU.

Format:

$$\begin{array}{llll} 1 & 14 & & \\ S & \text{CPUCF} & \left[\begin{Bmatrix}22\\ 8K\\ 16K\\ 32K\\ 64K\end{Bmatrix}\right] \left[\begin{Bmatrix}\text{STDSET}\\ \text{SCISET}\\ \text{COMSET}\\ \text{UNISET}\end{Bmatrix}\right] & [\text{DIRECT}]\quad [\text{PROTCT}]\end{array}$$

Subfields:

8K, 16K, 32K, or 64K = size of CPU, where K = 1,024 bytes

STDSET = standard instruction set

SCISET = scientific instruction set (also implies STDSET)

COMSET = commercial instruction set (also implies STDSET)

UNISET = universal instruction set (also implies SCISET, COMSET, and PROTCT)

DIRECT = direct control option

PROTCT = storage protect option

Notes:

1. The features specified describe the System/360 to be used as the object computer.

2. The subfields may be in any desired order.

3. Omission of the CPU size will give 64K, omission of the instruction set will give the standard set, and omission of DIRECT and PROTCT will suppress these options.

4. Omission of the CPUCF card, or presence of a card without a valid subfield, leads to assumption of 64K, UNISET, and DIRECT.

DUMP Card

Purpose: To specify control numbers for use in selecting the dynamic dumps to be executed (see Section 2).

Format:

```
1        14        22
S        DUMP      from 1 to 16 hexadecimal integers
```

Subfields: Each subfield may be a hexadecimal integer from 0 to F.

Notes:

1. If a dump request is specified in the object program and its control number does not match a control number on a DUMP card, the dump will not be executed.

2. If more than one DUMP card is used in a program, their effect is cumulative; i.e., all control numbers specified remain in effect until the end of the program.

FLOW Card

Purpose: To define the limits of a flow trace area (see Section 2).

Format:

```
1            14          22
S            FLOW        hhhh,hhhh
```

Subfields: hhhh = in each case, a hexadecimal address in simulated System/360 main storage, specifying, in order, the first and last bytes of the flow trace area.

Notes:

1. If the program is relocated, the addresses on the FLOW card will be relocated with the next segment loaded. If either address is not within the next segment, the card will be rejected. The addresses on the FLOW card will be treated as absolute if (a) the card is positioned after the last segment or (b) if loading is absolute or is accomplished by an IPL procedure or by a supervisor written in System/360 coding.

2. Up to ten FLOW cards may be used with each segment or program.

3. Once a flow trace area is defined, the definition remains in effect throughout the program. Subsequent FLOW cards may add to, but not reduce, the size of a flow trace area.

ICNT Card

Purposes:

1. To specify a limit to the number of machine instructions to be simulated, for use in protection against infinite loops.

2. To specify a count of instructions executed, by type.

Format:

```
1            14          22
S            ICNT        dddddd [TIME] [DETAIL]
```

Subfields: dddddd = a decimal integer of from 1-6 digits or a decimal integer of from 1-5 digits followed by either T or M, where T = 1,000 and M = 1,000,000.

TIME specifies that a count be kept of the number of times each type of machine instruction is executed (including completed and interrupted instructions) and causes the nonzero counts to be punched on-line in a form suitable for machine analysis (see Section 2). This subfield must not be used unless a 721 Card Punch is attached on-line as SYSPCH.

DETAIL specifies the same counts as TIME, but causes the nonzero counts to be printed at the end of the program.

Notes:

1. If an ICNT card is not provided, approximately 5 minutes of 7090 machine time will be allowed.

2. The decimal integer must be the first subfield specified; the other two may appear in any order.

3. If this card is placed within a supervisor program: (a) the TIME and DETAIL options, if specified, will apply to all subsequent programs, and (b) the count (dddddd) will apply to all subsequent programs unless another ICNT card occurs within the program.

INITL Card

Purpose: To specify the leftmost 40 bits of the initial program status word to be used when simulation of a program begins.

Format:

```
1          14         22
S          INITL      hhhhhhhhhh
```

Subfield: hhhhhhhhhh = 10 hexadecimal digits to be used as the leftmost 40 bits of the program status word; all 10 digits must be punched.

Note: This card is required only if loading is to be performed by the loader in the Support Package; in that case, if this card is omitted, the leftmost 40 bits of the initial PSW will be set to zeros.

IOCF Card

Purpose: To specify the input/output configuration of the System/360 and to specify which 7090 units are to be used in simulating the System/360 units.

Format:

```
1          14         22
S          IOCF       SYSxxx,hhh,type,[n,] options
```

Subfields: SYSxxx = name of 7090 system unit, which must be on Channel A or B if it is a 729 tape unit, and must be one of the following: SYSLB2, SYSLB3, SYSLB4, SYSIN1, SYSIN2, SYSOU1, SYSOU2, SYSPP1, SYSPP2, SYSCK1, SYSCK2, SYSUT1, SYSUT3, or SYSUT4; SYSUT2 is also allowed in SIMULATE actions and the field may be null if the type subfield is 1052; in the latter case, the SYSxxx subfield is ignored and SYSCRD and SYSPRT will be used to simulate the console keyboard and printer respectively.

hhh = System/360 input/output unit address, expressed as three hexadecimal digits.

type = System/360 unit type, which may be one of the following: 1052 (see note under SYSxxx above) , 1402R, 1402P, 1442R, 1442P, 1403, 1443, 2311, 2401 2402, 2403, or 2404. To simulate one 1442 for reading and punching, two IOCF cards are required; both must specify the same System/360 unit address, but one must specify 1442R (for reading) and the other 1442P (for punching) and each must specify a different 7090 unit. Both these 7090 units assigned to this 1442 must be on the same channel. The same is true when one 1402 is to be simulated, except that the two IOCF cards must specify different system/360 unit addresses.

n = decimal number less than 203. (Specified only for 2311.) Starting from cylinder n, as many cylinders will be simulated as can be fitted into the disk limits already assigned to IBSYS unit SYSxxx. Two 2311 cylinders are simulated on each 1301 (or 7320) cylinder. In the absence of this subfield, n is taken to be zero.

options = Options available for particular unit types, as follows:

L144	= 1443 with 144 print positions
7TRACK	= 2401, 2402, 2403, or 2404 with both seven-track and data converter features
7TNODC	= 2401, 2402, 2403, or 2404 with the seven-track feature only.
IMAGE	= 1442 with Card Image feature, or 1402 with Column Binary feature.

Notes:

1. A separate IOCF card is required for each unit to be simulated.

2. If the first subfield (SYSxxx) is invalid, the card will be ignored, except that if the type subfield is 1052, the contents of the SYSxxx subfield will be ignored.

3. Each IOCF card applies to all programs following it in the same action.

IPL Card

Purpose: To specify simulation of the Initial Program Loading procedure.

Format:

1	14	22
S	IPL	hhh

Subfield: hhh = a System/360 input/output unit address, expressed as from one to three hexadecimal digits.

Note: The IPL card causes simulation to begin immediately with an Initial Program Loading sequence from the specified unit; that unit must have been defined on an IOCF card. The deck structure required when using this procedure is given in Section 4.

OUTPUT Card

Purpose: To control the output from the Assembly Program.

Format:

```
1         14        22
S         OUTPUT    [NODECK]  [NOREFS]  [UNLIST]
```

Subfields:

NODECK	=	Suppress binary card deck production.
NOREFS	=	Suppress cross-reference table.
UNLIST	=	Suppress source program listing, except for those statements containing errors.

Note: The subfields may be in any desired order.

SUP Card

Purpose: To specify the mode in which Supervisor Call instructions are to operate.

Format:

```
1         14        22
S         SUP       1 to 4 subfields
```

Subfields: Each subfield must be a single decimal digit from 0-3 corresponding to an identification code used in the Supervisor Call instruction. It may be either unsigned or preceded by a minus sign. If unsigned, it specifies that subsequent Supervisor Call instructions with the same code are to operate in 7090 mode; if minus, they are to operate in System/360 mode. See Section 4 for details.

Note: For interrelationships of BASE and SUP cards, see Section 4.

TITLE Card

Purpose: To identify a card deck as a symbolic segment for assembly and to specify a title to be printed on the listing (see Section 1).

Format:

```
1         14        22
S         TITLE     page header for listing
```

Notes:

1. A TITLE card must precede every segment intended for assembly.

2. This card signals the 1401 Input Program to begin writing blocked records on the input tape.

END-OF-FILE CARD

Purpose: To define the end of a symbolic segment, and that of a program and its associated data, if any (see Section 3).

Format:

1 7 8	2 EOF

Note: Column 1 requires both 7 and 8 punches.

LOADER CARDS

This portion of Appendix B describes the formats of the cards produced as output by the 7090 Assembly Program and of two cards that may be used by the programmer in making modifications to the assembled output. It also describes the Load Terminate card, which must be supplied by the programmer for every program that requires simulation. Further details of the functions of the loader cards will be found in Section 1 and (in the case of the Load Terminate card) in Section 3.

OUTPUT CARDS

There are four principal types of output cards: (1) External Symbol Dictionary (ESD) cards, (2) Relocation List Dictionary (RLD) cards, (3) Text (TXT) cards, and (4) Load End (END) cards. ESD cards are produced in three different formats, depending on whether they contain program names, entry points, or external symbols.

The card code employed is a form of column binary that corresponds to the internal code of the System/360. This code is known as "IBM Extended Card Code." A table showing this code system is given in Appendix I. Certain fields in these cards, however, are punched in standard IBM card code. For example, the card identification symbols ESD, RLD, and TXT appear in standard card code in columns 2-4. For convenience, the explanation of each field either specifies actual, standard card code, or hexadecimal punching, or does not specify a code, in which case the extended card code punching applies.

External Symbol Dictionary Card (ESD)

Format 1 (Program Name)

CARD COLUMNS	CONTENTS
1	Loader card identification (12-2-9)
2-4	ESD (in standard card code)
5-10	(blank)
11-12	16 (12-0-1-8-9 and 12-11-1-8-9)
13-14	(blank)
15-16	External symbol identification
17-22	Program name (in standard card code)
23-24	(blank)
25	0 (in standard card code)
26-28	24-bit address of first byte of program as assembled
29	(blank)
30-32	Program length, in bytes
33-72	(blank)
73-76	Name field from TITLE instruction (see "Additional Facilities" in Section 1)
77-80	Sequence number

Format 2 (Entry Point)

CARD COLUMNS	CONTENTS
1	Loader card identification (12-2-9)
2-4	ESD (in standard card code)
5-10	(blank)
11-12	16 (12-0-1-8-9 and 12-11-1-8-9)
13-16	(blank)
17-22	Name of entry point (in standard card code)
23-24	(blank)
25	1 (in standard card code)
26-28	24-bit address of entry point as assembled
29-30	(blank)
31-32	External symbol identification of program name
33-72	(blank)
73-76	Name field from TITLE instruction
77-80	Sequence number

Format 3 (External Symbol)

CARD COLUMNS	CONTENTS
1	Loader card identification (12-2-9)
2-4	ESD (in standard card code)
5-10	(blank)
11-12	16 (12-0-1-8-9 and 12-11-1-8-9)
13-14	(blank)
15-16	External symbol identification
17-22	Name of external symbol (in standard card code)
23-24	(blank)
25	2 (in standard card code)
26-28	000 (three columns, each 12-0-1-8-9)
29-72	(blank)
73-76	Name field from TITLE instruction
77-80	Sequence number

Relocation List Dictionary Card (RLD)

CARD COLUMNS	CONTENTS
1	Loader card identification (12-2-9)
2-4	RLD (in standard card code)
5-10	(blank)
11-12	Number of bytes of information in variable field
13-16	(blank)
17-72	Variable field (see below)
73-76	Name field from TITLE instruction
77-80	Sequence number

Variable Field: The variable field consists of an integral number of RLD items. Each such item consists of the following fields:

1. Relocation header-i.e., the 2-byte External Symbol Identification of the symbol which determines the value of the load constant.

2. Position header-i.e., the 2-byte External Symbol Identification of the program segment in which the load constant occurs.

3. Flag-a 1-byte field containing four flag bits: a load constant length, bits 4 and 5; a complement flag, bit 6; and a continuation flag, bit 7.

4. Address-i.e., the 3-byte assembled address of the load constant.

For any RLD item, subfields 3 and 4 may be repeated as long as the continuation flag bit is on.

Text Card (TXT)

CARD COLUMNS	CONTENTS
1	Loader card identification (12-2-9)
2-4	TXT (in standard card code)
5	(blank)
6-8	24-bit address of first byte of text as assembled
9-10	(blank)
11-12	Number of bytes of text in card
13-14	(blank)
15-16	External symbol identification
17-72	Text (1 to 56 bytes, one per column)
73-76	Name field from TITLE instruction
77-80	Sequence number

Load End Card (END)

CARD COLUMNS	CONTENTS
1	Loader card identification (12-2-9)
2-4	END (in standard card code)
5	(blank)
6-8	24-bit address of an entry point to the object program as assembled (optional)
9-14	(blank)
15-16	External symbol identification
17-72	(blank)
73-76	Name field from TITLE instruction
77-80	Sequence number

MODIFICATION CARDS

Two types of cards are available to the programmer for making modifications to the assembled output of the 7090 Assembly Program. These are (1) Replace (REP) cards and (2) Include Segment (ICS) cards. The functions of these cards are described in Section 3. If these cards are used, they must be supplied by the programmer. For convenience, their format prescribes the use of standard IBM card code, and the cards may therefore be easily punched on a standard card punch. Their formats are as follows.

Replace Card (REP)

CARD COLUMNS	CONTENTS
1	Loader card identification (12-2-9)
2-4	REP (in standard card code)
5-6	(blank)
7-12	Assembled address of the first byte to be replaced (in hexadecimal)
13	(blank)
14-16	External symbol identification (in hexadecimal)
17-72	From 1 to 11 4-digit hexadecimal fields separated by commas (but no blanks), each replacing one previously loaded halfword
73-80	(not used)

Include Segment Card (ICS)

CARD COLUMNS	CONTENTS
1	Loader card identification (12-2-9)
2-4	ICS (in standard card code)
5-16	(blank)
17-22	Name of segment (in standard card code)
23	(blank)
24	, (comma)
25-28	Length in bytes of segment (in hexadecimal)
29-72	(blank)
73-80	(not used)

LOAD TERMINATE CARD (LDT)

As explained in Section 3, the Load Terminate card is used to indicate to the loader that the last segment of the current program has been loaded and that simulation is to begin. The card must be supplied by the programmer, using the following format, for any program to be simulated. Its position in the input deck is specified in Section 3.

CARD COLUMNS	CONTENTS
1	Loader card identification (12-2-9)
2-4	LDT (in standard card code)
5-16	(blank)
17-22	Name of symbolic entry point to the object program (in standard card code) (optional)
23-72	(blank)
73-80	(not used)

Following is a list of the status and sense information that will be supplied by the unit simulation routines for the 1052, 1402, 1442, 1403, 1443, 2311, 2401, 2402, 2403, and 2404. Channel and unit status information will be recorded in the channel status word (CSW) whenever a CSW is stored: sense information is stored in connection with normal input/output operations and can be made available for inspection by the use of a Sense instruction. Status and sense bits that are not listed below will never be set.

BIT			FUNCTION
Channel Status			
		0	Program-Controlled Interruption
		1	Incorrect Length
		2	Program Check
		3	Protection Check
1052 Printer-Keyboard			
	Unit Status:	0	Attention
		3	Busy
		4	Channel End
		5	Device End
		6	Unit Check
	Sense Byte:	0	Command Reject
1402 and 1442 Card Read Punch			
	Unit Status:	3	Busy
		4	Channel End
		5	Device End
		6	Unit Check
		7	Unit Exception
	Sense Byte:	0	Command Reject
		1	Intervention Required
		3	8-Bit Compare
		4	Invalid Card Code
		6	Unusual Command Sequence (1402 only)
1403 and 1443 Printers			
	Unit Status:	3	Busy
		4	Channel End
		5	Device End
		6	Unit Check
	Sense Byte:	0	Command Reject
2311 Disk Storage Drive			
	Unit Status:	1	Status Modifier
		3	Busy
		4	Channel End
		5	Device End
		6	Unit Check
		7	Unit Exception
	Sense Byte 1:	0	Command Reject
		1	Intervention Required
		3	Equipment Check
		4	Data Check

	6	Track Condition Check
	7	Seek Check
Sense Byte 2:	1	Track Overrun
	2	End of Cylinder
	3	Invalid Sequence
	4	No Record Found
	5	File Protected
	6	Missing Address Marker
Sense Byte 3:	0	Unsafe

2401, 2402, 2403, and 2404 Magnetic Tape Units

Unit Status:	3	Busy
	4	Channel End
	5	Device End
	6	Unit Check
	7	Unit Exception
Sense Byte 1:	0	Command Reject
	4	Data Check
Sense Byte 2:	1	Tape Unit Status A
	3	7-Track
	4	At Load Point
	5	Write Status
	7	Tape Indicator
Sense Byte 4:	0	R/W VRC

APPENDIX D: ASSEMBLY AND SIMULATION TIMING

ASSEMBLY TIMING

The speed of assembly will depend on the complexity of the program being assembled, the equipment available, and the input/output unit assignments. When operating in the ASSEMBLE mode, a typical program of 500 instructions or more will assemble at a rate of approximately 5,000 cards per minute, assuming (1) that the input has been prepared by the 1401 Input Program, (2) that a reasonably accurate card count has been given on a COUNT card, (3) that the computer used is a 7090 with standard 7090 System Monitor unit assignments, (4) that a tape density of 556 or 800 cpi is used on SYSIN1, SYSOU1, and SYSPP1, and (5) that a tape density of 800 cpi is used on SYSUT1 and SYSUT3.

The rate of assembly will be somewhat slower for programs operating in the ASSEMBLE,SIMULATE mode and also for programs that contain a high proportion of DC statements, especially those that involve floating-point conversions and/or duplication factors.

The rate of assembly will be somewhat faster if SYSIN1 and SYSOU1 are assigned to the same channel and SYSPP1, SYSUT1, and SYSUT3 to another.

SIMULATOR TIMING

The execution time for each instruction consists of the sum of the decoding time and the time taken by the individual instruction subroutine; these are analogous to the instruction fetch and execution times on System/360. The decoding time falls into a small number of individual cases that correspond to the instruction format, but the time taken for each instruction subroutine varies widely. In these circumstances, it is possible to give only an approximate estimate of simulation time.

It is estimated that the number of instructions simulated per second on the 7090 will range from 3,000 to 3,500, except in the case of decimal instructions. Use of the PROTCT, DETAIL, or TIME options will tend to reduce the average speed.

Decimal instructions are simulated at a rate drastically lower than that of other instructions.

No estimate of input/output timing is given other than to indicate that the corresponding 7090 input/output operations are overlapped with processing, while internal conversions and transmission between simulated System/360 main storage and 7090 buffers are not.

The loader in the Support Package will load relocatable programs at a rate of approximately 4,000 bytes per second.

APPENDIX E: 7090 OPERATION

The following text is a summary of information pertaining to the operation of a 7090 when using the Support Package.

7090 CONSOLE OPERATIONS

SENSE SWITCHES

The console sense switches may be used to initiate the procedures indicated in the following table:

SENSE SWITCH	PROCEDURE WHEN SWITCH IS ON
2	1052 Printer-Keyboard Attention interruption.
3	System/360 Console interruption.
4	External interruption.
5	Dump current program and skip remainder of present action.
6	Examine the other sense switches.

NOTE: Unless Sense Switch 6 is on, the other sense switches are inoperative. Sense switch 6 should be set last.

SENSE LIGHTS

The interpretation of 7090 sense lights is shown in the following table:

SENSE LIGHTS	INTERPRETATION
1	ON: IPL PROCEDURE IS IN PROGRESS.
2, 3, AND 4 TOGETHER	ON: CPU IS RUNNING (simulation of CPU Running Light). OFF: CPU is in WAIT status.

KEYS

The console keys are used in simulating the Direct Control Feature, as follows:

KEYS	USE
S	When on, cancels the Hold signal.
28-35	To enter a byte of data as input for use by a Read Direct instruction.
30-35	To enter an External Interruption Identification code.

MQ DISPLAY LIGHTS

The MQ Display Lights are used to display information relating to the direct control feature, as follows:

LIGHTS	INFORMATION DISPLAYED
S-7	R1 and R3 fields of a Write Direct or Read Direct instruction.
28-35	Byte produced as output by a Write Direct instruction.

ON-LINE PRINTER MESSAGES

All input messages entered from the 711 Card Reader in simulating input from the 1052 Printer-Keyboard will be printed on the 716 Printer on-line, preceded by the words CONSOLE INPUT.

All output messages printed on the 716 Printer in simulating output from the 1052 Printer-Keyboard will be printed on-line, preceded by the words CONSOLE OUTPUT.

All other messages are listed below. These will either request operator action, in which case the 7090 will pause, or will indicate a stage that has been reached in simulating a program.

ASSEMBLER

Disastrous - Assembly Terminated

A PERMANENT REDUNDANCY HAS OCCURRED ON UNIT XX FILE NO YYY BLOCK NO ZZZ.

XX is alphanumeric (e.g., A2) indication of the unit; YYY, ZZZ are decimal numbers. This message is followed by one of three subsidiary messages:

ASSEMBLY HAS BEEN TERMINATED.

TEN CONSECUTIVE REDUNDANCIES. ASSEMBLY HAS BEEN TERMINATED.

UNEXPLAINED REDUNDANCY ON TAPE USED FOR OUTPUT. ASSEMBLY HAS BEEN TERMINATED.

ASSEMBLER OR MACHINE ERROR. JOB SKIPPED. ERROR NO XXX.

XXX is a. decimal number indicating where in assembly the error occurs. It should be included on relevant Applied Programming Analysis Reports (APAR).

E.O.T. CONDITION ON UNIT. XX.
ASSEMBLY HAS BEEN TERMINATED.

XX is alphanumeric indication of the unit.

ICTL CARD ERROR. ASSEMBLY TERMINATED.

INTERNAL DICTIONARY FULL. REASSEMBLE IN SMALLER SEGMENTS. ASSEMBLY TERMINATED.

NAME TABLE FULL. REASSEMBLE IN SMALLER SEGMENTS. ASSEMBLY TERMINATED.

Serious - Simulation Deleted

A PERMANENT REDUNDANCY HAS OCCURRED ON UNIT XX.
FILE NO YYY BLOCK NO XXX.
RECORD NO VVV HAS BEEN DISCARDED.

VVV is a decimal number.

E.O.T. CONDITION ON UNIT XX
BINARY OUTPUT HAS BEEN DELETED.

ERROR DETECTED IN ABOVE ASSEMBLY. (SIMULATION DELETED).

Second half of message appears only if simulation was requested.

Possible Errors - Simulation Allowed

END OF FILE ENCOUNTERED BEFORE END CARD. ASSEMBLER PROCEEDS AS THOUGH END CARD DISCOVERED.

E.O.T. CONDITION ON UNIT XX
MOUNT NEW TAPE IF FURTHER OUTPUT REQUIRED
IF OUTPUT NOT REQUIRED THEN TO TERMINATE ASSEMBLY
DEPRESS SIGN KEY. IN BOTH CASES PRESS START TO CONTINUE.

POSSIBLE ERROR IN ABOVE ASSEMBLY. (SIMULATION FOLLOWS).

Other Messages

NO ERROR DETECTED IN ABOVE ASSEMBLY.

SIMULATOR

Messages Followed by 7090 Pause

LOAD CONSOLE INPUT INTO CARD READER.

Cards representing 1052 input should be readied in the 711 Card Reader and START pressed. If input is not available, switching on the S key and pressing START will cause the current program to be dumped and the remainder of the action to be skipped.

PUT NEW TAPE ON UNIT XX

Alphanumeric indication of the unit is given in case end of tape is recognized during simulation.

READ DIRECT. IF NO INSTRUCTIONS GIVEN WITH PROGRAM OR ABOVE ENSURE THAT S KEY IS UP AND PRESS START.

Causes cancellation of hold signal without data entry.

TURN OFF SWITCHES 2 TO 6.

The Sense Switches have been checked and found to be set at the start of an action, or after the switches have been used.

WAIT STATE. IF NO INSTRUCTIONS GIVEN WITH PROGRAM OR ABOVE, DEPRESS SENSE SWITCHES 5 AND 6 AND PRESS START.

Current program will be dumped and remainder of action skipped.

WRITE DIRECT. IF NO INSTRUCTIONS GIVEN WITH PROGRAM OR ABOVE ENSURE THAT S KEY IS UP AND PRESS START.

Causes cancellation of hold signal without data entry.

Message Not Followed by 7090 Pause

SIMULATION TERMINATED BY SWITCHES 5 AND 6.

APPENDIX F: 1401 OPERATION

The 1401 Input Program is provided as a self-loading card deck. A card deck conforming to the structure outlined in Section 3 can be used to prepare a system input tape for the 7090. This tape is mounted on Tape Unit 1. Sense Switch A and the I/O Check Stop switch must be ON.

Tape Rewind

The program does not rewind Tape 1 at the beginning or the end of the job. Hence a composite tape can be prepared, if required, using any other suitable 1401 card-to-tape program to prepare other special sections.

Errors

After four unsuccessful attempts to write tape, Tape Write Redundancy causes a halt and the printing of this message:

```
TAPE ERROR.  PRESS START TO TRY AGAIN.
```

Up to four attempts to write are made each time the start key is pressed.

The following two messages, each of which is followed by a halt, indicate end of processing by the 1401 Input Program. The first message calls attention to an unsuccessful completion of processing.

```
END TAPE.  PUT CURRENT JOB ON FRESH TAPE.
```

```
END OF JOB.  OUTPUT ON TAPE UNIT 1.
```

When "last card" is read, or end of tape is encountered, the program writes a tape mark, a $STOP card image, and another tape mark. Three backspace commands then are issued. Thus, unless end of tape has been encountered, more input can be processed and the tape marks and $STOP will be overwritten.

A Card Reader Validity Check halt will be caused if a card in a symbolic source deck contains a non-BCD code. In this case, the job must be abandoned and returned to the programmer.

Restart Procedure

To avoid reloading the program when several tapes are to be prepared, restart manually from location 2580.

Timing

The program processes System/360 source decks at approximately 700 cards a minute, and other cards at 400 a minute.

The following text lists all Support Package off-line messages, with any necessary explanation.

ASSEMBLER

Disastrous--Assembly Terminated

ASSEMBLER OR MACHINE ERROR. JOB SKIPPED. ERROR NO XXX.

XXX is a decimal number indicating where in Assembly the error occurs. It should be included on relevant Applied Programming Analysis Reports (APAR).

ICTL CARD ERROR. ASSEMBLY TERMINATED.

INTERNAL DICTIONARY FULL. REASSEMBLE IN SMALLER SEGMENTS. ASSEMBLY TERMINATED.

NAME TABLE FULL. REASSEMBLE IN SMALLER SEGMENTS. ASSEMBLY TERMINATED.

Serious--Simulation Deleted

ERROR DETECTED IN ABOVE ASSEMBLY. (SIMULATION DELETED).

Second half of message appears only if simulation was requested.

Possible Errors--Simulation Allowed

COUNT CARD IS MISSING.

COUNT CARD VARIABLE FIELD IS INVALID.

Count card is ignored.

DUPLICATE COUNT CARD. PREVIOUS CARD IGNORED.

END OF FILE ENCOUNTERED BEFORE END CARD. ASSEMBLER PROCEEDS AS THOUGH END CARD DISCOVERED. ASSEMBLER HAS GENERATED A LOADER END CARD.

POSSIBLE ERROR IN ABOVE ASSEMBLY. (SIMULATION FOLLOWS).

Other Messages

NO ERROR DETECTED IN ABOVE ASSEMBLY.

SIMULATOR

Supervisor

ABNORMAL TERMINATION. REMAINDER OF ACTION IS OMITTED.

This message follows termination for any of these causes: setting on Sense Switches 5 and 6, an SVC 2 call, or an input/output error.

ERROR ON IOCF CARD. SIMULATION DELETED.

FLOW CARD IGNORED. ILLEGAL ADDRESS.

NO MORE ROOM FOR IOCF TABLES. SIMULATION DELETED.

REDUNDANCY ON INPUT TAPE. CARD IGNORED.

REMARK.

Added to print of a card image. This card is not recognized as a Support Package control card.

SIMULATION TERMINATED BY SWITCHES 5 AND 6.

The operator has set Sense Switches 5 and 6 to cause a dump of the current program and skipping of the remainder of the action.

UNIT NOT ASSIGNED, NOT ATTACHED OR WRONG TYPE ERROR ON IOCF CARD. SIMULATION DELETED.

Input/Output

ERROR IN IPL.

ERROR IN IPL UNIT NOT FOUND.

NO FURTHER CONSOLE INPUT SUPPLIED.

TAPE BUFFER SIZE EXCEEDED.

In all four cases above the current program is dumped, the first message under "Supervisor" above is printed, and the remainder of the action is skipped.

MOUNT ANOTHER TAPE. UNIT XXXXXX.

Occurs when end of tape is encountered. XXXXXX is the octal address of the tape unit.

Read Direct

THE OPERATOR HAS SET XX TO BE READ INTO YYYYYY.

XX is a byte in hexadecimal, YYYYYY is the address in hexadecimal.

Dynamic Dump

INVALID C. LIST

An invalid address has been given in a control list; this entry is skipped.

INVALID FORMAT.

A non-existent dump format has been called for; the dump request is ignored.

Program Interruptions

```
                          ⎧ INVALID OPERATION.
                          ⎪ PRIVILEGED OPERATION.
                          ⎪ INVALID EXECUTE.
PROGRAM INTERRUPT DUE TO  ⎨ PROTECTED ADDRESS.
                          ⎪ INVALID ADDRESS.
                          ⎪ INVALID SPECIFICATION.
                          ⎩ INVALID DATA.
```

INSTRUCTION ADDRESS IS XXXXXX.

The address is given in hexadecimal.

Loader

LOADER DIAGNOSTIC. ADDRESS IS OUTSIDE SIMULATED STORAGE AVAILABLE. CARD NUMBER XXXXXX.

LOADER DIAGNOSTIC. BCD CARD FOUND IN INPUT TO LOADER. CARD NUMBER XXXXXX.

LOADER DIAGNOSTIC. CHARACTER IN REPLACE OR ICS CARD IS NOT HEXADECIMAL. CARD NUMBER XXXXXX.

LOADER DIAGNOSTIC. ENTRY POINT ****** HAS TWO DEFINITIONS. CARD NUMBER XXXXXX.

****** represents a six-character symbolic name.

LOADER DIAGNOSTIC. EXTERNAL SYMBOL DICTIONARY CARD FOLLOWS TEXT. CARD NUMBER XXXXXX.

LOADER DIAGNOSTIC. ILLEGAL CARD COLUMN IN INPUT TO LOADER. CARD NUMBER XXXXXX.

LOADER DIAGNOSTIC. ILLEGAL CARD TYPE IN LOADER INPUT. CARD NUMBER XXXXXX.

LOADER DIAGNOSTIC. INSUFFICIENT SPACE AVAILABLE FOR LOADER TABLES. CARD NUMBER XXXXXX.

LOADER DIAGNOSTIC. INSUFFICIENT STORAGE AVAILABLE FOR THIS PROGRAM. CARD NUMBER XXXXXX.

LOADER DIAGNOSTIC. LOAD TERMINATE CARD NOT PRECEDED BY END CARD. CARD NUMBER XXXXXX.

If any of the above messages is given, loading is completed but simulation is deleted.

LOADER DIAGNOSTIC. WARNING--BYTE COUNT ON TEXT CARD EXCEEDS 56. CARD NUMBER XXXXXX.

LOADER DIAGNOSTIC. WARNING--ADDRESS IS OUTSIDE RANGE OF PROGRAM SEGMENT. CARD NUMBER XXXXXX.

LOADER DIAGNOSTIC. WARNING--CARD COLUMN 1 DOES NOT CONTAIN LOADER ID. CARD NUMBER XXXXXX.

LOADER DIAGNOSTIC. WARNING--END OF FILE FOUND BEFORE LOAD TERMINATE CARD. CARD NUMBER XXXXXX.

LOADER DIAGNOSTIC. WARNING-- ****** IS DEFINED WITH TWO LENGTHS. CARD NUMBER XXXXXX.

****** represents a six-character symbolic name.

LOADER DIAGNOSTIC. WARNING--SECTION ****** HAS NO REAL DEFINITION. CARD NUMBER XXXXXX.

****** represents a six-character symbolic name.

LOADER DIAGNOSTIC. WARNING--TEXT FOLLOWS REPLACE OR RLD CARDS. CARD NUMBER XXXXXX.

LOADER DIAGNOSTIC. WARNING-- ****** USED AS ENTRY AND PROGRAM SEGMENT. CARD NUMBER XXXXXX.

****** represents a six-character symbolic name.

Error conditions are flagged and diagnosed as explained below.

Warning Errors

E More than 14 external symbols specified. The offending symbols are ignored.

G Range error in floating point DC statements E and D. The result is set to zero.

H A storage address in a machine instruction is not aligned to the correct boundary; or,

A register does not have the correct specification--i. e., odd register in some double length instructions, odd or oversize register in floating-point instructions.

L The label of a statement is illegal in some way, and is not used. This flag also appears if a label appears in a statement which itself is not used.

M The label of a statement has been given multiple definitions. If this symbol does not appear in the variable field of any other statement, only a warning is given.

P An assembler instruction may be illegal in some way. A presumed result is given.

Q A subfield that must be explicitly stated is missing in the variable field. This applies to machine instructions only, and the offending field is taken as zero.

T One or more subfields in the variable field of the statement have values too large. The value is truncated to the left.

Z Column 72 is non-blank.

Serious Errors

A A subfield of the variable field specified as relocatable (e.g., address field in a USING statement) has been given an absolute value. It is ignored.

B A START, ENTRY, or EXTRN card is out of order. The statement is ignored.

C The location counter has overflowed the limit of 2 to the 24th power minus 1. The least significant 24 bits are taken, and the flag is shown against the offending statement.

D Error in DC or DS assembler instruction specification; the statement is abandoned.

F Error in format in machine operation, or in a DC or DS statement. Variable field ignored.

I A storage address subject to using table look-up cannot be mapped into base and displacement. The variable field is set to zero.

M A symbol with multiple definition is in the variable field; the field is set to zero or ignored as in U below.

N This assembler instruction is invalid in some way and is ignored. This flag will normally be shown with another flag more closely defining the reason for failure.

O The operation code mnemonic is not in the mnemonic table. The whole statement is ignored.

R Relocation error found in subfield expression(s). Examples of such errors include:

Negative relocatable value.
Multi-relocation.
Negative relocation.
Illegal relocatable value in absolute field.

S An illegal symbol has occurred in variable field. An illegal symbol is defined as:

Commencing with 0-9.
Containing an illegal character, e.g., $.
More than six characters in length.

U A symbol appearing in the variable field is undefined. The field is set to zero in machine instructions, and the entire statement is ignored in assembler instructions.

V Symbol appearing in variable field of ORG or EQU statement has not been previously defined. The statement is ignored.

X An external symbol has been used erroneously; e.g., in an ORG assembler instruction, in an expression involving another relocatable symbol, etc.

APPENDIX I: IBM SYSTEM/360 EXTENDED CARD CODE

The following table shows the card codes that correspond to the System/360 internal codes. Each card code occupies one card column and represents one eight-bit internal code occupying one byte of main storage.

The card codes are arranged in a numerical sequence based on the order of the rows in the card. The sequence begins with the 12-punch and is followed by the 11-punch and the punches from 0 through 9. The 12-punch is represented in the table by the letter T and the 11-punch by the letter E.

The table shows, first, the card code, then the internal code, the corresponding hexadecimal coding, and (where applicable) the character that may be represented by the code. In actual use within the computer, a code may represent either a numeric (binary) value or an actual character, depending on the program. In either case, the same internal code is represented by the same card code.

CARD CODE	INTERNAL CODE	HEX CODE	CHARACTER
(blank)	0100 0000	40	(blank)
T	0101 0000	50	&
TE	0110 1010	6A	
TE0	0111 0000	70	
TE01	1011 0001	B1	
TE018	1011 0000	B0	
TE0189	0011 0000	30	
TE019	0111 0001	71	
TE02	1011 0010	B2	
TE028	1011 1010	BA	
TE0289	1111 1010	FA	
TE029	0111 0010	72	
TE03	1011 0011	B3	
TE038	1011 1011	BB	
TE0389	1111 1011	FB	
TE039	0111 0011	73	
TE04	1011 0100	B4	
TE048	1011 1100	BC	
TE0489	1111 1100	FC	
TE049	0111 0100	74	
TE05	1011 0101	B5	
TE058	1011 1101	BD	
TE0589	1111 1101	FD	
TE059	0111 0101	75	
TE06	1011 0110	B6	
TE068	1011 1110	BE	
TE0689	1111 1110	FE	
TE069	0111 0110	76	
TE07	1011 0111	B7	
TE078	1011 1111	BF	
TE0789	1111 1111	FF	
TE079	0111 0111	77	
TE08	1011 1000	B8	
TE089	0111 1000	78	
TE09	1011 1001	B9	
TE1	1001 0001	91	
TE18	1001 0000	90	
TE189	0001 0000	10	
TE19	0101 0001	51	
TE2	1001 0010	92	
TE28	1001 1010	9A	
TE289	1101 1010	DA	
TE29	0101 0010	52	
TE3	1001 0011	93	
TE38	1001 1011	9B	
TE389	1101 1011	DB	
TE39	0101 0011	53	
TE4	1001 0100	94	
TE48	1001 1100	9C	
TE489	1101 1100	DC	
TE49	0101 0100	54	
TE5	1001 0101	95	
TE58	1001 1101	9D	
TE589	1101 1101	DD	
TE59	0101 0101	55	
TE6	1001 0110	96	
TE68	1001 1110	9E	
TE689	1101 1110	DE	
TE69	0101 0110	56	
TE7	1001 0111	97	
TE78	1001 1111	9F	
TE789	1101 1111	DF	
TE79	0101 0111	57	
TE8	1001 1000	98	
TE89	0101 1000	58	
TE9	1001 1001	99	
T0	1100 0000	C0	
T01	1000 0001	81	
T018	1000 0000	80	
T0189	0000 0000	00	
T019	0100 0001	41	
T02	1000 0010	82	
T028	1000 1010	8A	
T0289	1100 1010	CA	
T029	0100 0010	42	
T03	1000 0011	83	
T038	1000 1011	8B	
T0389	1100 1011	CB	
T039	0100 0011	43	
T04	1000 0100	84	
T048	1000 1100	8C	
T0489	1100 1100	CC	
T049	0100 0100	44	
T05	1000 0101	85	

CARD CODE	INTERNAL CODE	HEX CODE	CHARACTER
T058	1000 1101	8D	
T0589	1100 1101	CD	
T059	0100 0101	45	
T06	1000 0110	86	
T068	1000 1110	8E	
T0689	1100 1110	CE	
T069	0100 0110	46	
T07	1000 0111	87	
T078	1000 1111	8F	
T0789	1100 1111	CF	
T079	0100 0111	47	
T08	1000 1000	88	
T089	0100 1000	48	
T09	1000 1001	89	
T1	1100 0001	C1	A
T18	0100 1001	49	
T189	0000 1001	09	
T19	0000 0001	01	
T2	1100 0010	C2	B
T28	0100 1010	4A	
T289	0000 1010	0A	
T29	0000 0010	02	
T3	1100 0011	C3	C
T38	0100 1011	4B	. (period)
T389	0000 1011	0B	
T39	0000 0011	03	
T4	1100 0100	C4	D
T48	0100 1100	4C	
T489	0000 1100	0C	
T49	0000 0100	04	
T5	1100 0101	C5	E
T58	0100 1101	4D	(
T589	0000 1101	0D	
T59	0000 0101	05	
T6	1100 0110	C6	F
T68	0100 1110	4E	+
T689	0000 1110	0E	
T69	0000 0110	06	
T7	1100 0111	C7	G
T78	0100 1111	4F	
T789	0000 1111	0F	
T79	0000 0111	07	
T8	1100 1000	C8	H
T89	0000 1000	08	
T9	1100 1001	C9	I
E	0110 0000	60	- (minus)
E0	1101 0000	D0	
E01	1010 0001	A1	
E018	1010 0000	A0	
E0189	0010 0000	20	
E019	1110 0001	E1	
E02	1010 0010	A2	
E028	1010 1010	AA	
E0289	1110 1010	EA	
E029	0110 0010	62	
E03	1010 0011	A3	
E038	1010 1011	AB	
E0389	1110 1011	EB	
E039	0110 0011	63	
E04	1010 0100	A4	
E048	1010 1100	AC	
E0489	1110 1100	EC	
E049	0110 0100	64	
E05	1010 0101	A5	
E058	1010 1101	AD	
E0589	1110 1101	ED	
E059	0110 0101	65	
E06	1010 0110	A6	
E068	1010 1110	AE	
E0689	1110 1110	EE	
E069	0110 0110	66	
E07	1010 0111	A7	
E078	1010 1111	AF	
E0789	1110 1111	EF	
E079	0110 0111	67	
E08	1010 1000	A8	
E089	0110 1000	68	
E09	1010 1001	A9	
E1	1101 0001	D1	J
E18	0101 1001	59	
E189	0001 1001	19	
E19	0001 0001	11	
E2	1101 0010	D2	K
E28	0101 1010	5A	
E289	0001 1010	1A	
E29	0001 0010	12	
E3	1101 0011	D3	L
E38	0101 1011	5B	$
E389	0001 1011	1B	
E39	0001 0011	13	
E4	1101 0100	D4	M
E48	0101 1100	5C	*
E489	0001 1100	1C	
E49	0001 0100	14	
E5	1101 0101	D5	N
E58	0101 1101	5D	)
E589	0001 1101	1D	
E59	0001 0101	15	
E6	1101 0110	D6	O (letter)
E68	0101 1110	5E	
E689	0001 1110	1E	
E69	0001 0110	16	
E7	1101 0111	D7	P
E78	0101 1111	5F	
E789	0001 1111	1F	
E79	0001 0111	17	
E8	1101 1000	D8	Q
E89	0001 1000	18	
E9	1101 1001	D9	R
0	1111 0000	F0	0 (zero)
01	0110 0001	61	/
018	0110 1001	69	
0189	0010 1001	29	
019	0010 0001	21	
02	1110 0010	E2	S
028	1110 0000	E0	
0289	0010 1010	2A	
029	0010 0010	22	
03	1110 0011	E3	T
038	0110 1011	6B	, (comma)
0389	0010 1011	2B	
039	0010 0011	23	
04	1110 0100	E4	U
048	0110 1100	6C	%
0489	0010 1100	2C	
049	0010 0100	24	
05	1110 0101	E5	V
058	0110 1101	6D	

CARD CODE	INTERNAL CODE	HEX CODE	CHARACTER
0589	0010 1101	2D	
059	0010 0101	25	
06	1110 0110	E6	W
068	0110 1110	6E	
0689	0010 1110	2E	
069	0010 0110	26	
07	1110 0111	E7	X
078	0110 1111	6F	
0789	0010 1111	2F	
079	0010 0111	27	
08	1110 1000	E8	Y
089	0010 1000	28	
09	1110 1001	E9	Z
1	1111 0001	F1	1
18	0111 1001	79	
189	0011 1001	39	
19	0011 0001	31	
2	1111 0010	F2	2
28	0111 1010	7A	
289	0011 1010	3A	
29	0011 0010	32	
3	1111 0011	F3	3
38	0111 1011	7B	#
389	0011 1011	3B	
39	0011 0011	33	
4	1111 0100	F4	4
48	0111 1100	7C	@
489	0011 1100	3C	
49	0011 0100	34	
5	1111 0101	F5	5
58	0111 1101	7D	' (quotation mark)
589	0011 1101	3D	
59	0011 0101	35	
6	1111 0110	F6	6
68	0111 1110	7E	=
689	0011 1110	3E	
69	0011 0110	36	
7	1111 0111	F7	7
78	0111 1111	7F	
789	0011 1111	3F	
79	0011 0111	37	
8	1111 1000	F8	8
89	0011 1000	38	
9	1111 1001	F9	9

INDEX

CUT ALONG LINE

READERS' COMMENTS

IBM 7090/7094 Support Package for System/360
Form C28-6501-2

Your comments regarding this publication will help us improve future editions. Please comment on the usefulness and readability of the publication, suggest additions and deletions, and list specific errors and omissions.

USEFULNESS AND READABILITY

fold fold

SUGGESTED ADDITIONS AND DELETIONS

ERRORS AND OMISSIONS (give page numbers)

fold fold

Name ______________________

Title or Position ______________________

Address ______________________

FOLD ON TWO LINES, STAPLE AND MAIL
No Postage Necessary if Mailed in U. S. A.

STAPLE

FOLD

FOLD

FIRST CLASS
PERMIT NO. 81

POUGHKEEPSIE, N.Y.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY

IBM CORPORATION
P.O. BOX 390
POUGHKEEPSIE, N.Y. 12602

ATTN: PROGRAMMING SYSTEMS PUBLICATIONS,
DEPT. D58

CUT ALONG LINE

FOLD

FOLD

STAPLE

IBM®

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601